ControlTower™ Console Management for Linux User's Guide

Release 3L

Part Number: 15-10126-00, Version A
Revision Date: June, 2005

Limitation of Liability

Information contained in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty.

Carlo Gavazzi Computing Solutions makes no warranty, expressed or implied, with respect to this manual and any related items, their quality, performance, merchantability, or fitness for any particular use. It is solely the purchaser's responsibility to determine its suitability for any particular use.

In the interest of improving internal design, operational function, and/or reliability, Carlo Gavazzi Computing Solutions reserves the right to make changes to the products described in this document without notice. No guarantee, express or implied, is made that products of Carlo Gavazzi Computing Solutions will be compatible with future versions of the hardware systems and operating systems specified in this manual. YOU MUST READ THE SOFTWARE LICENSE AGREEMENT IN THE BACK OF THIS MANUAL AND RETURN THE PRODUCT UNOPENED IF YOU DO NOT AGREE TO BE BOUND BY ITS CONDITIONS.

Trademarks

Carlo Gavazzi Computing Solutions, Carlo Gavazzi, the Carlo Gavazzi Computing Solutions logotype, the Carlo Gavazzi logotype, Apollo Multiport, Nova Multiport, Aries Multiport, ControlTower, Explorer Multiport, LANMultiServer, Saturn Multiport, SBox, Vanguard Multiport, WANMultiServer, XP7 Expansion Chassis, XP-7R Rack-Mounted Expansion Chassis, XP-SB Expansion Chassis are trademarks of Carlo Gavazzi Computing Solutions.

SPARC is a registered trademark of SPARC International, Inc. in the United States and other countries.

CARLO GAVAZZI
COMPUTING SOLUTIONS

Contents

CARLO GAVAZZI
COMPUTING SOLUTIONS

CHAPTER 1

About this Book

Introduction

The *ControlTower Console Management for Linux User's Guide* describes how to install, configure and use ControlTower software. It also provides reference information.

Document Organization

This manual is organized as follows:

Chapter 1, About this Book	Describes target audience, conventions, related manuals for this document and registration information for ControlTower.
Chapter 2, About ControlTower™ Console Management System for Linux	Describes ControlTower product, system components, and sample application.

Chapter 3, Getting Started	Describes information for site preparation, hardware drivers, license keys, and managed devices for ControlTower.
Chapter 4, Installing ControlTower Software	Describes how to install ControlTower.
Chapter 5, Security and Configuration Concepts	Provides security and advanced configuration concepts.
Chapter 6, Configuring ControlTower	Provides configuration instructions using the Command Line Interface.
Chapter 7, Administering Managed Devices	Describes how to administrate managed devices using ControlTower.
Chapter 8, “Warranty & Maintenance Information.”	Describes product Warranty information.
Appendix A, “Command Reference.”	Lists man pages of ControlTower related User Commands.
Appendix B, “Default Configuration File.”	Lists the default configuration file.
Appendix C An Example Configuration	Shows examples of a LOCAL file, a group file, and device files.
Glossary Terms & Definitions	Presents frequently used terms and definitions.

Who Should Use This Book

This book is a user’s guide and reference for System Administrators who are using ControlTower to manage servers.

Document Conventions

Table 1 describes the symbolic conventions used in this guide.

TABLE 1. Conventions

Symbol	Description
screen display	Graphic text that appears on screens, menus and dialog boxes appears in sans serif font.
User input	User input values appear in **boldface**. These are characters or commands you type literally.
emphasis	Italics are used in the text for emphasis, titles, and variables.
	This caution symbol marks notes about possible damage to computer equipment or data if a procedure or process isn't followed according to instructions.
	This warning symbol marks notes about possible electrical shock to yourself or electro-static damage to your equipment unless you follow special instructions.
	This symbol marks special text passages that contain additional information such as notes you should know about or tips you should consider when installing, operating, or maintaining this product.

Related Manuals

For more information, refer to the following manuals:

- Your Carlo Gavazzi Computing Solutions Aurora brand Multiport Serial Controller User's Manual
- Your Linux distribution documentation
- On-line man pages

Getting Help

If you need to reach us, you can contact us by

- The Web: **www.gavazzi-computing.com** for product literature, phone numbers and address.
- Phone service: 508-588-6110 Mon–Fri, 8:30AM–5:00 PM Eastern Time. To expedite service, have your product serial number and your system information available.
- FAX: 508-588-0498. Attn: Customer Service and Support
- Email: **support@gavazzi-computing.com**

Product Registration

To receive standard warranty coverage on your product, including 90 days of free technical support, you must print, fill out, and fax or mail back the Warranty Registration Card that is located in Chapter 8, “Warranty & Maintenance Information.” Phone support can only be provided after product registration is complete. Extended Hardware and Software Support Agreements can be purchased to provide additional coverage.

Sending in this card also lets us keep you up-to-date on the complete line of Carlo Gavazzi Computing Solutions’ products.

If you have any questions or comments on your product, contact our Customer Service and Support Department at **support@gavazzi-computing.com** or our Sales Department at **sales@gavazzi-computing.com**.

CARLO GAVAZZI
COMPUTING SOLUTIONS

CHAPTER 2

About ControlTower™ Console Management System for Linux

What is ControlTower?

ControlTower™ Console Management System provides a reliable time and cost saving solution for monitoring and controlling multiple devices through an RJ-45/ DB-25 interface from a central location or by remote access. It enables a single Linux based system to function as a common console (monitor and keyboard) for managed devices. The ControlTower System is available for a PCI bus multiport serial controller.

System Components

ControlTower Console Management System consists of both software and hardware components. ControlTower Software, consisting of Server and Viewer Client packages, resides on a Linux based x86 system. The ControlTower Host System provides a common console and maintains system logs for all managed devices. No additional software is required on the managed devices.

ControlTower Viewer Client software, in addition to residing on the Host, may reside on multiple systems that have network or modem capabilities to the ControlTower Host System. Any function that can be performed from a managed devices's

keyboard and display can be performed remotely from a ControlTower Viewer Client, including monitoring log files, running diagnostics, and rebooting managed devices.

Sample Application

You can use the ControlTower Viewer to access one or more ControlTower Hosts via the network, enabling you to monitor and administer any number of systems in any number of locations from a single, central location, or from any number of locations you choose, as is shown in Figure 1.

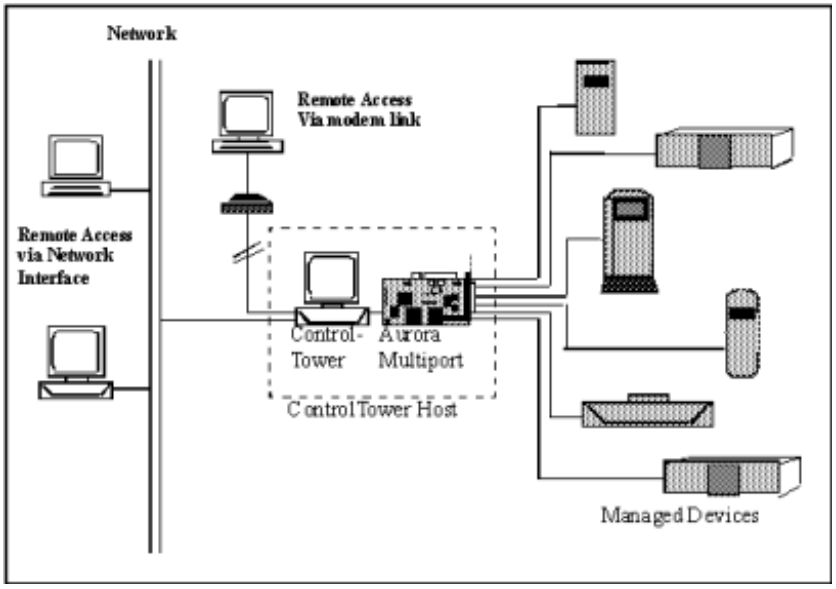

FIGURE 1. Console Management with ControlTower

CARLO GAVAZZI
COMPUTING SOLUTIONS

CHAPTER 3

Getting Started

Introduction

This chapter describes steps you must take before installing ControlTower software. It tells how to select an appropriate ControlTower Host and how to properly identify managed devices. It also lists materials you need for installation.

Before Installing

Before installing the ControlTower software you must complete the following tasks:

- Get the ControlTower package for your distribution
- Install/update the server package
- Install the license key
- Install the client application

Under Linux, most package management systems will allow you to simply upgrade the package. No need to uninstall. In all cases, this allows you to maintain your configuration.

Instructions on these tasks are found in this chapter. When these tasks are complete, you can proceed with installation.

Select the Host Machine

You can use either a PCI Bus Linux x86 machine as the ControlTower Host. The machine you choose must meet the following minimum requirements:

PCI Systems

Host:	Linux x86 system
Memory:	128 MB RAM (minimum)
Operating System:	Mandriva, RedHat, SuSe, and generic Linux distributions
Serial Controller	Aries Multiport™ 8000P or 16000P
Hardware:	XP-7R™ or PCI Expansion Chassis
Disk Space:	5 MB free in `/usr` ; 50 MB free in `/var`

The indicated memory requirements are based on the assumption that ControlTower software is run on a dedicated server.

Important Host Selection and Set-up Considerations

Your ControlTower Host is a critical component of your console management solution. Carlo Gavazzi Computing Solutions recommends the following steps to increase the security, availability and performance of your ControlTower Host:

- The ControlTower Host system should be a dedicated system. It should not be used by applications or users that might interfere with its console management role.
- The host machine should be attached to a UPS (uninterruptible power supply) of sufficient capacity to ensure that it will be up as long or longer than all managed devices.
- The host should not depend on NFS-mounted disks for its operation.

- The host should not depend on NIS (Yellow Pages) or NIS+ for its operation.
- Minimize the number of user accounts.
- Minimize host access, both physically and through the network (via filters/firewalls.)
- The command “host id” must return a non-nul hexadecimal string
- The server needs a network interface.

Break Signal Considerations

The supported Aurora brand multiport serial hardware from Carlo Gavazzi Computing Solutions has been thoroughly tested to verify that it does not transmit spurious break signals. Nevertheless, Carlo Gavazzi Computing Solutions recommends that you take the following precautions:

- Attach all DB25 connectors with screws, and ensure that all RJ45 connectors are firmly latched.
- Avoid disconnecting and reconnecting the network connection on running systems.
- Avoid power-cycling the ControlTower Host at times when managed device operation is critical.
- If it is necessary to stop ControlTower processes, use `/etc/init.d/cmgrd stop`.
- After connecting (or reconnecting) a managed device console port to the ControlTower Host, verify that the managed device is operational by connecting using a Viewer Client. See the `cmgr(1)` man page for further information.
- Verify operation of *all* systems after power-cycling the ControlTower Host or reloading the Aurora brand Multiport Serial Driver.
- See the `kbd(1)` man page for information on how to enable/disable break on the console serial port.
- Attach your host machine to a UPS (uninterruptible power supply) of sufficient capacity to ensure that it will be up as long or longer than all managed devices.

The break signal mostly affects Sun servers in their default configurations.

Identify Managed Devices

ControlTower allows you to manage devices which have an RS-232 console port. Systems other than those running Sun Solaris must be tested for compatibility with ControlTower.

Managed Devices Worksheet

Complete the Managed Device Worksheet (page 3-4) to help plan the types of devices you will be managing with ControlTower. Some examples are provided. Photocopy the worksheet for additional managed devices.

TABLE 2. Managed Devices Worksheet

Port #	Managed Device Name	Device Type	Console Port Connector Type	DCE or DTE
		(modems, printers, workstations, or servers)	(DB25, DB9, RJ45)	

Verify Materials

Before installing ControlTower, verify that you have all necessary materials. They are listed in the following hardware and software charts:

TABLE 3. Hardware Parts List

Qty.	Description
1	Dedicated server host—enter host ID#_____
*var	User's Manuals for Linux based system
1	Multiport Serial Controller Hardware
1	Serial Controller Card User's Manual with Device Driver CD-ROM
1	Driver Release Notes
1	Distribution cable or Breakout Box
1	Serial Test Plug
*var	Adapters for Managed Devices (optional)

TABLE 4. Software Parts List

Qty.	Description
1	ControlTower CD ROM—enter serial#_____
1	ControlTower User's Guide
1	ControlTower Extended Support Agreement

*var=Variable Quantity--depends on situation

Install New Hardware and Drivers

Install new Aurora brand hardware on your chosen ControlTower Host system before you begin the ControlTower software installation. For information on installing the hardware, see the Carlo Gavazzi Computing Solutions user's guide for the hardware you are installing.

CECS does not provide support for third-party hardware. Any serial hardware that supports standard Term I/O works with this version of ControlTower.

Obtain License Key File

ControlTower requires a license key file for correct operation.

To obtain a license key, please contact Carlo Gavazzi Computing Solutions Customer Service and Support. The product serial number and license information will be posted on the inside of the CD case. Contact information is as follows:

- The Web: **`www.gavazzi-computing.com`** for product literature, phone numbers and address.
- Phone service: From US exchanges 508-588-6110 Mon–Fri, 8:30AM–5:00PM Eastern Time. To expedite service, have your product serial number and your system information available.
- FAX: 508-588-0498; Attn: Customer Service and Support
- Email: **`support@gavazzi-computing.com`**

Telephone numbers occasionally change. Please see web site for current contact information.

When you contact Customer Service and Support you'll need to provide:

- your ControlTower serial number. **(???? Need to clarify for Linux)**
- the `hostid` of the system on which you have installed ControlTower.

Set Up Managed Devices

Connect Managed Devices to Host

Whether or not you are installing new Aurora brand hardware, you will need to connect managed devices to the ControlTower Host via asynchronous null modem cables. You can also use a straight through cable and asynchronous null modem cable adapters.

Use one of the cable connections shown in Figure 2 and Figure 3 (or a straight cable with an asynchronous null modem adapter) to connect managed devices to the Aurora brand hardware. For additional pinouts, contact Customer Support at Carlo Gavazzi Computing Solutions. (Contact information is found in the section "Obtain License Key File" on page 3-6).

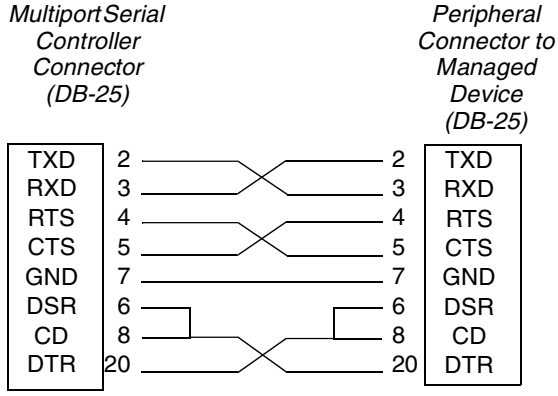

FIGURE 2. Asynchronous DB-25-to-DB-25 Null Modem Cable (XON/XOFF Handshaking)

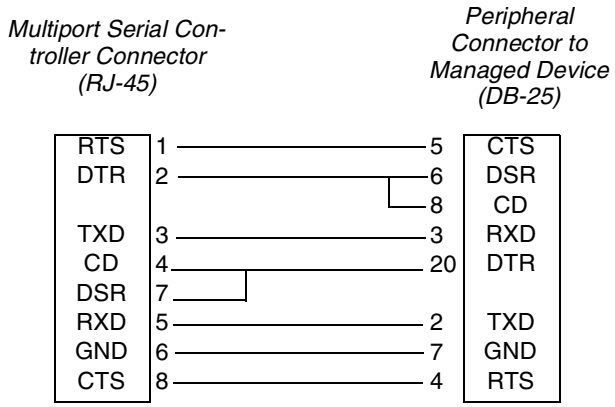

FIGURE 3. Asynchronous RJ-45-DB-25 Null Modem Adapter (Out-of-Band Flow Control)

CARLO GAVAZZI
COMPUTING SOLUTIONS

CHAPTER 4

Installing ControlTower Software

Introduction

This chapter tells how to install ControlTower software. Prior to installation, you must complete the steps in Chapter 3, Getting Started.

Installation includes several tasks:

- Check to see if a previous version of ControlTower is installed.
- Mount the CD-ROM
- Add packages
- Install the License Key
- Install ControlTower Software on remote systems

These tasks are described in this chapter.

Checking to See if a Previous Version of ControlTower is Installed

To check for existing ControlTower software:

1. Log in as root:

   ```
   login: root
   Password: <root_password>
   ```

2. Check for existing ControlTower software by typing:

   ```
   system#  rpm | cmgrd
   ```

   This command will output CMGRD-3.03-x if the package is installed.

Installing New ControlTower Software

To install ControlTower software, you will need a host that is equipped with a CD-ROM drive.

If the host does not have a CD-ROM drive, you will need to install the software through another machine on the network that does or download it from the website. Contact Carlo Gavazzi Computing Solutions Customer Service and Support for instructions on installing ControlTower software over a network.

Files and Directories

ControlTower software is installed in the following directories:

/usr/sbin	The server binaries
/usr/bin	The client binaries
/etc/AURAcmgr	Configuration directory
/etc/logrotate.d	Log rotation
/etc/pam.d	Authentification
/etc/init.d	Sart/Stop script
/usr/share/man/	Manual pages
/var/log/AURAcmgr	Log files directory

Adding a Package

On SuSE, Mandriva, and Red Hat, the package is installed by:

rpm -Uhv <RPM file>

In some cases, you may have to do a:

rpm -Uhv --no-deps <RPM file>

On Gentoo, you have two files:

cmgrd-<version>.tgz

cmgrd-<version>-portage.tgz

You need to:

1 - Setup a portage overlay directory in make.conf. If you don't have one defined you can do:

echo "PORTDIR_OVERLAY=/usr/local/portage'>> /etc/make.conf"

2 - Go to the portage overlay directory (create it if needed and unpack the "portage" file)

cd /usr/local portage

tar xfz cmgrd-<version>-portage.tgz

3 - Copy the "source" file to your portage "distfiles" directory

cp cmgrd-<version>.tgz /usr/portage/distfiles

4 - Emerge the software

emerge cmgrd

CARLO GAVAZZI
COMPUTING SOLUTIONS

CHAPTER 5

Security and Configuration Concepts

This chapter presents important ControlTower security issues. It also provides information on how to configure:

- Encryption
- Log file management
- User-access to ControlTower servers
- Error logging
- Compatibility with previous versions

Configuration Information

All ControlTower parameters are applied in a hierarchy depending on where the parameters are set. Parameters set in the LOCAL or DEFAULT files at the top level (`/etc/AURAcmgr/`) apply to all managed devices unless overridden by settings at a lower level. Parameters set in a group configuration file (`/etc/AURAcmgr/`***<group>***`/`***<group>***`.grp`) override settings at the top level and device configuration files (`/etc/AURAcmgr/`***<device>***`.cfg` or `/etc/AURAcmgr/`***<group>***`/`***<device>***`.cfg`) override

group and top level settings. For more information refer to “Configuring Groups” on page 6-4.

Parameter settings only override parameters of the same name (except for the `device` and `stty` settings which are transparent). For instance, logdir set in a device file will override the logdir setting of the `LOCAL` file. However, there are parameters that interact with parameters of a different name and these each have their own hierarchy. Examples of this will be described as they are encountered.

See Appendix B, “Default Configuration File.” , for a complete listing of the `DEFAULT` file. Also see the `config(4)` man page.

Security Information

Since ControlTower sessions may involve the use of the root password, or may involve root access on a managed device or remote communications between the Viewer the ControlTower Host, you will want to keep security issues in mind when setting up and maintaining ControlTower.

Remote Access Security

This version of ControlTower supports encryption of communications between a Viewer client running on a remote system and the ControlTower Host. This feature mitigates the security risks of transmitting sensitive data over TCP/IP networks.

Carlo Gavazzi Computing Solutions recommends that you *always* enable encryption when using the Viewer remotely, unless your TCP/IP connection to the server is over a secure LAN environment. You can also use SSH to encrypt a remote connection to a Viewer running on the ControlTower Host.

To enable encryption for a managed system, include the line

```
DefaultEncrypt=128
```

in its configuration file. As an alternative, you can add this line to the LOCAL file to enable encryption for all managed systems by default.

The line

```
ForceEncrypt=true
```

will cause any requests for remote connections that do not support encryption at the default level to be refused by the ControlTower Host.

Also, if encryption is not enabled for a managed system, the Viewer client can enable it on connect (with V. 3.0 Hosts only) using the command

```
cmgr -f 128 <server_name>
```

Other ControlTower Security Features

ControlTower 3L uses PAM to control access to the server. In it's default configuration, all of the users that can log onto the server system can access the server. By default they are granted "read only" access. Access rights are controlled by the auth users in /etc/AURAcmgr.

Log File Management

You can configure a number of aspects of log file management, including:

- Where
- Content
- Timestamps
- Protection

Storage Directory for Log Files

Note that under Linux, log file rotation is done using the "logrotate: application." If you change the location of the log files, you will have to adapt:

/etc/logrotate.d/cmgrd

The name of the log file is:

AURAcmgr-<service name>.log or AURAcmgr-<group name>-<service name.log

You must take care when defining your groups and services that there is no name clash. For example, having Group/1-port20.cfg and Group-1/

port20.cfg would create a clash for the log files AURAcmgr-Group-1-port20.log.

Log File Rotation

Log file rotation uses “logrotate.”

/etc/logrotate.d/cmgrd defines the rotation parameters. See the logrotate manual for more details.

Log File Timestamping

Time stamps are periodically placed into the log files. The timestamp frequency is determined by the `logstamp` parameter. The default setting is 60 minutes. Periodic timestamp format is determined by the `logstamp-format` parameter. For more information about `logstampformat` see “logstampformat” on page 6-9.

In addition, each log entry begins with a timestamp. This can be turned off by setting `loglinestamp` to null (`loglinestamp=`). You can control the appearance of the time stamp by changing the format characters. See also logstamp, logstampformat and loglinestamp on page 6-9.

Log File Protections

The default protection mode, owner, and group for a managed device’s log-files are as follows:

```
logmode=u=rw
logowner=root
loggroup=sys
```

Values you can specify for these are as follows:

Protection Mode
The value specified for `logmode` can be expressed either as an octal number (e.g., 600), or as a comma-separated sequence of absolute modes strings (e.g., `u=r,o=rw`). See the `chmod` (1) man page for a detailed description of these possible values.

Owner
The value specified for `logowner` can be expressed as a decimal user-id, or as a username from the password database.

Group
The value of `loggroup` can be expressed as a decimal group-id, or as a group name from the groups database.

Log files may contain sensitive system information (including passwords). You should carefully consider to whom you make them accessible. Through the use of Regular Expressions in the log filter, sensitive information may be removed. Refer to log filtering on page 5-5.

Log Filtering

Log filtering selects which lines of information are written to the log file based on sequences of characters found within the line using Regular Expression matching. Log filtering is configured using the `logfilter` parameter to specify a file name and populating a file of that name with Regular Expression commands that will ‘drop’ (or ‘keep’) lines that would otherwise be written (or not) to the log file. This can conserve disk space.

Authorization Parameters

You can configure a number of aspects of user-access to the ControlTower Host, including:

- Whether and how local-domain access to a ControlTower Host is permitted
- Whether, and from what hosts, remote TCP/IP access to the ControlTower Host is permitted
- The usernames to use for remote access to a ControlTower Host
- Setting permissions for access to managed devices

Local Access Control

UNIX-domain access is used for local Command Line Viewer Client access. This is the case when the **CONSOLE_SERVERS** environment variable is not set, and the Command Line Viewer is started without specifying a remote server (***<device_name>@<server_name>***).

Use the `localenable` parameter to permit or deny local-domain access to ControlTower Hosts. By default, local-domain access is permitted (`localenable=true`).

Use the `localauth` parameter to specify whether users must enter a password for local access to ControlTower Hosts. By default, password entry is not required
(`localauth=false`).

Remote (TCP/IP) Access Control

Use the `tcpenable` parameter to permit or deny access via TCP/IP to ControlTower Hosts. By default, TCP/IP access is permitted (`tcpenable=true`).

TCP/IP access can be controlled on a system-by-system basis by entering the IP addresses of servers into `tcpallow` and `tcpdeny` in a comma-delimited list.

Username for Remote Access

- All remote connections over a network require entry of a password. This password may be the same for all devices managed by a ControlTower Host using the `authuser` parameter. Alternatively, authorization can be managed through the `authfile` parameter. This is the recommended method for authorization since it gives much better control over access.
- All network connections are checked using IP address access control lists to permit or deny connections from specific hosts or entire networks (or net blocks).
- The configuration parameters for TCP/IP network connections are the following:

  `tcpenable`: enable use of TCP/IP connections

`authuser`: name of the only user with access to the ControlTower server. Ignored if `authfile is` set.

`authfile`: name of a list of authorized users and their permissions.

- To use network client access using `authuser`, the user specified in the `authuser` parameter must be a valid account.
- To use network client access using `authfile`, set the parameter to the name of a file containing users and their permissions. This file name can contain an absolute path, or if a path is not given, the file is expected to be in the directory containing the .cfg file specifying this file name. Each entry in the `authfile` file should have a valid account name. Valid account names are specified by your system administrator. The account need not have a usable shell. (i.e., use `/usr/bin/false`.)

If you enable network access (tcpenable=true, authuser=**<user_ with_ password>**, or authfile= **<file_with_list_of_users>**, you should consider setting up network access control lists using tcpallow and tcpdeny. See the config(4) man page.

User Permissions to Access Managed Devices

Using the `authfile` parameter is the recommended method for controlling access to managed devices. Using `authfile`, security can be configured so that each user has different permissions for each managed device and different sets of users can have access to different devices or sets of devices (groups).

Error Logging

All messages output by the ControlTower Host program that runs for each device (`conserv`) are sent to `syslog` tagged with the `daemon` facility code (except for security-related messages, which are tagged with the `auth` facility code.) See the `syslogd(1M)` man page for information on configuring the syslog daemon.

If you are having difficulty using `syslog` to debug problems, contact Customer Service and Support. See “Getting Help” on page 1-4.

CARLO GAVAZZI
COMPUTING SOLUTIONS

CHAPTER 6

Configuring ControlTower

Introduction

This chapter tells how to configure ControlTower software using the Command Line Interface (CLI). This includes how to set up configuration files for each managed device.

This chapter assumes strong knowledge of UNIX™ commands. If any listed commands are unknown or their usage is unclear, please see the man page for the command (`man` ***<command>***).

Use of ControlTower software involves important security issues. Be sure to read Chapter CHAPTER 5, Security and Configuration Concepts.

Configuration Tasks

Configuration of the ControlTower software consists of the following tasks:

- Set up managed device configuration files

- Set up the environment

Set Up Managed Device Configuration Files

The default configuration for all devices managed by a ControlTower server is specified in the `DEFAULT` configuration file in `/etc/AURAcmgr/`.

In addition, each device is represented by a configuration file in the format ***<managed_device_name>***`.cfg`. This configuration file can override the configuration specified in the `DEFAULT` file.

The name you give this configuration file is the name by which the managed device will be known to ControlTower. It is recommended that the configuration file for the managed device have the same name as the managed device. Configuration file names:

- may be from 1 to 64 characters long
- may be the same as the network name, but are not required to be
- must have the extension ".`cfg`" (or ".`grp`" for group configuration files).
- must reside in `/ett/AURAcmgr/` or a group directory directly under this directory.
- may not begin with a period "."

The configuration file for a managed device must contain, at a minimum, the console server device pathname specifying the server port to which the managed device console port has been connected. For example:

```
device=/dev/ttyAxx for Linux kernel 2.6.x
device=/dev/cuaxx   for kernel 2.4.x
```

For a managed device to belong to a group, its configuration file must be located in the group subdirectory under `/etc/AURAcmgr/`. The subdirectory must contain a".`grp`" file with the same name as the subdirectory. The ".`grp`" file may be empty or contain parameters that will be applied to all devices in the group. The group file may not contain the device parameter.

You must create a configuration file for each managed device.

Creating a Configuration File for a Managed Device

Perform the following procedure to create a minimal configuration file for a managed device:

To create a configuration file

Log in as root (or use `su`):

```
login: root
Password: <root_password>
```

1. cd to /etc/AURAcmgr/:

```
system# cd /etc/AURAcmgr/
```

2. Using the text editor of your choice (`vi` is shown here), create a file having the name by which you wish this managed device to be known:

```
system# vi <managed_device_name>.cfg
```

   The file must have a `.cfg` extension.

3. Insert into the file the line `device=/dev/ttyy[AORS]xxx/<port_number>` where <*port_number*> is the port to which this device has been attached. For example:

```
device=/dev/ttyy[AORS]128
```

   for a managed device connected to serial port 128.

When you have created configuration files for all managed devices connected to the server, you are ready to start ControlTower.

Configuration File Hierarchies and Precedence

Configuration file hierarchies are illustrated in Figure 4. The leaf nodes override anything above. For example, Group overrides `LOCAL`, and `LOCAL` overrides `DEFAULT`, but a configuration file for a managed device overrides all of these for that device.

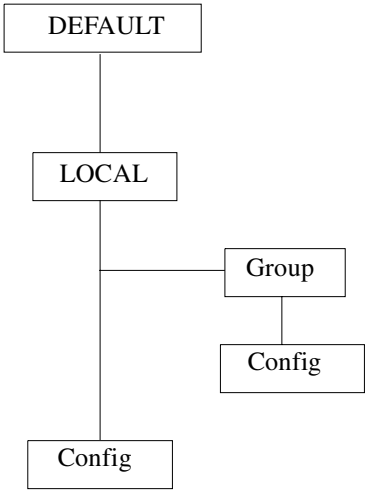

FIGURE 4. Configuration File Hierarchies

Configuring Groups

You can associate groups of managed devices using a subdirectory. Each subdirectory must have a file with the same name as the subdirectory and the extension `.grp`. This file contains the group configuration parameters. All devices that have configuration files within this directory will have the group configuration file parameters applied to them unless these parameters are set in the devices' individual configuration files.

To avoid confusion, it is recommended that configuration file names be unique across groups. The configuration file name cannot be the same as the group file name.

Creating Logins For Remote Users

You may want to perform one or more of the following tasks to set up the environment on the ControlTower server system:

If you intend to allow remote access to this ControlTower Host system, you must create user accounts for all users who are authorized to use ControlTower remotely. If you want separate logins for each user having access to the ControlTower Host system remotely, use the `authfile` parameter and create a separate login for each user listed in the `authfile` file. See "Username for Remote Access" on page 5-6. If you want to associate only one user account with any or all managed devices, set the `authuser` parameter to a user name and create a user account with the name specified. The default `authuser` name defined in the `DEFAULT` configuration file is "auracmgr". You can redefine this for all managed devices in the `LOCAL` file, or for an individual managed device in its ***<managed_device_name>***.cfg file.

To create the "auracmgr" user:

```
system# useradd -s /usr/bin/true [-u <user_id>] auracmgr
    system# passwd auracmgr
    system# New Password: <auracmgr_password>
    system# Re-enter New Password: <auracmgr_password>
```

Starting the ControlTower Server Software

To start the ControlTower server, type:

```
/etc/init.d/cmgrd  start
```

You can start individual services with the command:

```
/etc/init.d/cmgrd start <service name> (this does not work on
Gentoo).  Use /usr/sbin/conserv <service name> instead.
```

You can start all of the services in a group (except for Gentoo) with this command:

/etc/init.d/cmgrd start <group name>

If your service names end in a number not starting with 0 (zero), you can start ranges of services. For example, port12.cfg is good, but port01.cfg is not.

/etc/init.d/cmgrd start <service prefix><start>-<end>

E.g.

/etc/init.d/cmgrd start port14-18 will start services with configuration files port14.cfg, port15.cfg, port16.cfg, port17.cfg, port18.cfg.

More than one service, group, and/or range can be started at the same time:

E.g.

/etc/init.d/cmgrd start port14-18 port 20 Group 1/port33 Group2

Before starting the server, you must complete the installation tasks described in Chapter 4, "Installing ControlTower Software".

During installation, the ControlTower server start/stop scripts are placed in `/etc/init.d` to allow it to be started when the system is booted. You must use your system's admin tools to include cmgrd in your boot process.

Stopping the ControlTower Server Software

To stop all ControlTower Server Software processes

To stop the ControlTower server, type:

```
/etc/init.d/cmgrd stop
```

You can stop individual services with the command:

```
/etc/init.d/cmgrd stop <service name> (this does not work on
Gentoo). Use /usr/sbin/conserv <service name> instead.
```

You can stop all of the services in a group (except for Gentoo) with this command:

/etc/init.d/cmgrd stop <group name>

If your service names end in a number not starting with 0 (zero), you can stop ranges of services. For example, port12.cfg is good, but port01.cfg is not.

/etc/init.d/cmgrd start <service prefix><stop>-<end>

E.g.

/etc/init.d/cmgrd stop port14-18 will start services with configuration files port14.cfg, port15.cfg, port16.cfg, port17.cfg, port18.cfg.

More than one service, group, and/or range can be stopped at the same time:

E.g.

/etc/init.d/cmgrd stop port14-18 port 20 Group 1/port33 Group2.

Configuration Parameters and Defaults

The following are the configuration parameters for ControlTower:

exclusive

`exclusive` gives ControlTower sole access to a given port. The default is `true`. If this parameter is set to `false`, other programs can open this port. This is NOT recommended.

uulock

`uulock` sets up a uucp-compatible lock file so that other programs do not use the port to send data to another system. The default is `true`.

stty

`stty` controls serial port parameters. The default value is **`9600 cs8 -crtscts -cstopb -parenb -parext -parodd -ixoff -ixon istrip`**. Permissions for `stty` are set in the configuration file with `ttychanges`. See the `stty` man page for `stty` options and other information.

ttychanges

`ttychanges` allows Viewer Client programs to change tty line parameters. The default is `true`.

breakstring

`breakstring` allows you to configure what is sent instead of a break signal. If the `breakstring` parameter is not set, the break action will send a break signal to the managed device. If the `breakstring` parameter has been configured, the specified text will be sent. The default is null (`breakstring=`). If set, to unset this parameter in a configuration file at a lower level, set it to `*novalue*`. `breakstring` may contain backslash-escaped characters: \r \n \t \ooo (one or more octal digits) \xXX (two hex digits).

logdir

`logdir` allows you to specify a directory to which log files will be written. The default is `/var/log/cmgrlog`. The value must be expressed as an absolute path. If the managed device is a member of a group, the device log file will be created in a subdirectory with the same name as the group.

logfile

`logfile` allows you to explicitly specify the file to which log output will be written. `logfile` defaults to `logdir/<servername>`, but can be customized to a pathname for each server individually or all servers combined. If multiple server outputs are combined, it is recommended that you disable `logstamp`.

lognameprepend

If enabled, `lognameprepend` prepends the server name to all logs made. Useful for combining several server log outputs to one file.

loginput

If `loginput` is set to `true`, all text that is entered into the Viewer Client will be written to the log file, including passwords. The default is `false`.

logmessages

`logmessages` controls whether messages generated by ControlTower are written to the log file. The default is `true`.

logstamp

`logstamp` inserts a line containing a time stamp into the log file at regular intervals which you determine. Valid intervals are 10, 20, 30, or 60 minutes. A value of 0 means no logstamp is written. The default value is `60`.

logstampformat

`logstampformat` contains the format of the time stamp that is inserted into the log file. See the `strftime(3C)` man page for valid format variables. The default is `**********%c**********`.

loglinestamp

`loglinestamp` specifies that a time and date stamp will be written on each log line received from a managed device. If `loglinestamp` is null, no line-by-line timestamping will be performed. The default is `%c`. See the `strftime(3C)` man page for valid format variables.

logmode

`logmode` specifies the log file permissions mode. The default is `u=rw`, which means that only the owner has read/write access to the log files. The available values are `ugoa=rwx`. Different permissions can be set for different users (user, group, or other) by listing the different users and their permissions separated by a comma, for instance, `u=rwx,g=rw,o=r`. See the `chmod` (1) man page for more information.

logowner

`logowner` specifies the owner of the log files. This would be the “u” in the description of `logmode`. The default value is `root`. Users are listed in `/etc/passwd`.

loggroup

`loggroup` specifies the group to which the owner of the log files belongs. This would be the “g” in the description of `logmode`. The default value is `sys`. Defined groups are listed in `/etc/group`.

logfilter

`logfilter` specifies the name of a file that contains commands that drop or keep lines in the log file based on Regular Expressions. The name of the file may include an absolute or relative path. If relative, the path is relative to the directory in which the `logfilter` parameter is set. There is no default value. To unset `logfilter` in a configuration file at a lower level, set it to `*novalue*`. See the `regex(3)` man page for information on Regular Expressions.

The available commands that filter log file lines using Regular Expressions, are `keep` and `drop`.

The rules of log filtering are as follows:

- Each line of data from the managed device is tested in turn against each regular expression starting from the top of the list.
- When a match is found, processing stops. Therefore only the action of the first match is performed.
- If no match is found, the default action of `keep` occurs.

The following examples shown in Table 5 work collaboratively:

TABLE 5. Filtering Examples

Regular Expression	Application
keep /Mary had a little lamb/	**All lines containing the text "Mary had a little lamb" will be logged.**
drop /lamb/	**All other lines containing the term "lamb" will be excluded from the log file.**
# Comment	**All text that begins with a # is a comment and is ignored.**

Regular Expressions

(Review section thoroughly)

Certain characters have special meaning in Regular Expressions. The most common are listed below along with their usage.

- $ - the end of a string

- ^ - the beginning of a string, or NOT if it occurs at the beginning of (a) character(s) in square brackets
- . - any single character other than a newline
- + - one or more occurrences of the preceding character, e.g., a+
- *- zero or more occurrences of the preceding character, e.g., a*
- () - delimits individual characters that form a string
- [] - delimits a set of characters which must contain every character in the string for a match, '-' denotes a range of characters, e.g., [a-z]
- \ - if before any of the special characters above, makes that character represent itself

Here are some examples:

```
^Mary[a-z]*lamb$
```

matches any string with 'Mary' at the beginning, any number (including 0) of lower case letters and spaces, and 'lamb' at the end.

```
^[^0-9]+$
```

matches any string that doesn't have at least one digit.

```
(has).
```

matches any string with at least one occurrence of 'has' with at least one character after it.

It is important that `logfilter` files keep and drop the data they are expected to. To verify that they do, a syntax checker has been supplied that can run a `logfilter` file against sample input. The syntax checker is `filtertest` in `/opt/AURAcmgr/sbin`. The syntax is `/opt/AURAcmgr/sbin/filtertest` ***<filterfile>*** [***<inputfile>***].

If the ***<filterfile>*** contains a `drop` command and the first example of a Regular Expression (**`drop /^Mary[a-z]*lamb$/`**) and the ***<inputfile>*** contains:

```
Mary had a little lamb.
Mary had a little lamb
Mary had 9 lambs
Mary has a little lamb
```

The output from `filtertest` will be:

```
KEEP: Mary had a little lamb.
DROP: Mary had a little lamb
KEEP: Mary had 9 lambs
DROP: Mary has a little lamb
SUMMARY: keep: 2, drop: 2
```

If the ***<filterfile>*** contains a `drop` command and the second example of a Regular Expression (**`drop /^[^0-9]+$/`**), the output from `filtertest` will be:

```
DROP: Mary had a little lamb.
DROP: Mary had a little lamb
KEEP: Mary had 9 lambs
DROP: Mary has a little lamb
SUMMARY: keep: 1, drop: 3
```

If the ***<filterfile>*** contains a `drop` command and the third example of a Regular Expression (**`drop /(has)./`**), the output from `filtertest` will be:

```
KEEP: Mary had a little lamb.
KEEP: Mary had a little lamb
KEEP: Mary had 9 lambs
DROP: Mary has a little lamb
SUMMARY: keep: 3, drop: 1
```

If you would prefer to use a delimiter other than '/', any character can be used as long as it begins and ends the Regular Expression.

authuser

`authuser` specifies a user who is authorized to use a particular port. If `authuser` is used instead of `authfile`, there will only be one authorized user per port, so everyone who needs access to this port will use the same user name and password. The default is `auracmgr`.

Either `authuser` or `authfile` can be used for each device. Both cannot be used simultaneously.

authfile

`authfile` is set to the name of a file that contains a comma-separated list of users and their permissions. The file name specified by `authfile` can include an absolute or relative path. If relative, the path is relative to the directory in which the `authfile` parameter is set. This parameter is unset by default. Once set, to unset `authfile` in a configuration file at a lower level, set it to `*novalue*`.

The permissions that can be assigned to users are listed below. Text in the parentheses are the parameters as seen in the `authfile` file.

- Attach (`attach`)--The user can acquire read/write permission for the managed device if there is currently no other user in read/write mode for that managed device. If another user is attached, and the user with `attach` permission tries to attach, the user will be attached in read-only mode.
- Force Attach (`fattach`)--A user who has this permission can acquire read/write permission to a device even if there is another user attached. Another user who is attached is forced into read-only mode.
- stty (`stty`)--The user has permission to set stty parameters for devices.
- Break (`break`)--the user has permission to send a break string to the managed device
- None (`none`)--The user has no authority to do anything.
- All (`all`)--the user has all of the above permissions.

Permissions can be combined with a plus (+) sign or subtracted with a minus (-) sign. The following is a sample `authfile` file with multiple users:

```
# This file contains users who have access
# to the devices in group
auracmgr attach+fattach+break
developer1 all-stty
# The following user will be allowed view-only sessions
developer2 none
```

tcpenable

`tcpenable` determines whether remote machines are allowed to connect to the server using TCP/IP over a network. The default value is `true`.

tcpallow

`tcpallow` contains a list of machines that are allowed to connect to the server using TCP/IP over a network. If set, `tcpallow` will contain a comma-delimited list of IP addresses or host names, either of which can be followed by a mask. There is no default value. If set, to unset `tcpallow` in a configuration file at a lower level, set it to `*novalue*`.

tcpdeny

`tcpdeny` contains a list of machines that are not allowed to connect to the server using TCP/IP over a network. The syntax is the same as for `tcpallow`. There is no default value. If set, to unset `tcpdeny` in a configuration file at a lower level, set it to `*novalue*`.

defaultencrypt

`defaultencrypt` enables Twofish encryption over TCP/IP connections. The default is 0. Acceptable values are 0, 128 and 256. This only takes effect if the client on a local machine connects to this server over TCP/IP using the CONSOLE_SERVERS environment variable.

forceencrypt

`forceencrypt` causes all incoming TCP/IP connections to be dropped unless they accept the `defaultencrypt` or greater encryption level. This will also cause all v2.0 and v1.0 TCP client connections to be dropped.

localenable (formerly UNIXenable)

`localenable` determines whether the command line viewer (`cmgr`) has access to the local ControlTower Server Host. The default is `true`.

localauth (formerly UNIXauth)

`localauth` controls whether a password is required when using the command line viewer from the ControlTower Server Host. The default is `false`.

disconnectidle

`disconnectidle` sets the maximum amount of time, in minutes, that the Viewer Client session is allowed to remain idle, regardless of whether it is in read-only or read/write mode. After this point, the Viewer Client is disconnected. If the parameter is set to 0, there will be no automatic disconnect. The default value is `0`.

detachidle

`detachidle` sets the maximum amount of time, in minutes, that the Viewer Client is allowed to remain idle while in read/write mode. After this point, the Viewer Client is set to read-only mode. If the parameter is set to 0, there will be no forced shift into read-only mode. The default value is `0`.

autoresp

`autoresp` can be set to one of three values: "vt100 DC1", “vt100 DC2” or “vt100 DS”. If set, the server will send adequate response to the vt100 queries, even when no client is connected. This can be used to insure that some PC console are active since they rely on the vt100 sequence during boot to determine if they are connected and, if so, the baudrate to use.

CARLO GAVAZZI
COMPUTING SOLUTIONS

CHAPTER 7

Administering Managed Devices

Using Command Line Interface for Managed Devices

This chapter tells how to administer and monitor managed devices using the Command Line Interface (CLI). You can administer and monitor managed devices through the ControlTower Host and from remote Viewer Clients.

This chapter assumes knowledge of UNIX commands. If any listed commands are unknown or their usage is unclear, please see the man page for the command (`man <command>`).

Setting the CONSOLE_SERVERS Variable

Set the **CONSOLE_SERVERS** environment variable.

If you will be running ControlTower on multiple servers connected in a network and would like to use the Viewer Client to monitor devices managed by a different server, you may want to set **CONSOLE_SERVERS** to specify these servers. If **CONSOLE_SERVERS** contains a comma-separated list of ControlTower servers, the Viewer Client will have access to all of the listed servers.

Set **CONSOLE_SERVERS** as follows:

ksh or sh:

```
CONSOLE_SERVERS=<server_system1>,<server_system2>,...
export CONSOLE_SERVERS
```

csh or tcsh:

setenv CONSOLE_SERVERS <*server_system1*>, <*server_system2*>,...

The **CONSOLE_SERVERS** environment variable only exists for the server on which it was set. Each ControlTower server from which you wish to connect to devices on other servers should have the **CONSOLE_SERVERS** environment variable set.

About CLI Viewer Client

ControlTower CLI Viewer Client is a user interface to the ControlTower server software. After contacting the ControlTower server, the Viewer Client establishes an active session with the console port of *a single device* managed by that server. You must run one instance of Viewer Client software for each device you want to view.

There is only one client command:

cmgr

CLI Viewer Client Operation

When you run the Viewer Client, you can specify in the command line the managed device you want to view, and the access mode.

Specifying a Managed Device to View

To connect to a managed device, the Viewer Client needs to know:

- the name of the managed device, and

- the name of the ControlTower Host machine that manages that device, if it is on a remote server.

These are specified in the command line when you run the Viewer Client, as follows:

```
system# cmgr <managed_device_name>[@<host_name>]
```

If you do not specify **@**<***host_name***>, the Viewer Client looks in the **CONSOLE_SERVERS** environment variable for a comma-separated list of systems running ControlTower servers. If **CONSOLE_SERVERS** is not set, Viewer Client defaults to the local ControlTower server.

If you do not specify <***managed_device_name***>, the Viewer Client displays a list of devices accessible from the local server.

Specifying the Access Mode

By default, the Viewer Client connects to the managed device in read-only mode. In read-only mode, you must enter an escape sequence to send input to the managed device console port.

You can, however, specify that the Viewer Client should `attach` to the managed device console port (i.e., read-write mode) with the ~a or ~A escape sequences. (For more information, see "Escape Sequences" on page 7-6.) In read/write mode, the Viewer Client window functions as a console terminal attached to the managed device. The users and permissions listed in the `authfile` file determine which escape sequences are available to which user. For more information on `authfile`, see "authfile" on page 6-13.

Command Examples

These command examples show how to use the ControlTower Viewer Client to view managed devices. The **CONSOLE_SERVERS** environment variable determines how devices on local and remote servers are specified for viewing. The following commands have the described effects when **CONSOLE_SERVERS** is not set.

- List devices managed by the local server

```
system# cmgr
cmgr: must have system name
hercules apollo ulysses agamemnon
[cmgr viewer exiting]
```

- View a device managed by the local server in read-only mode:

```
system# cmgr hercules
```

If security is being administered using `authuser`, the next line entered will be:

`password:<authuser_password>`

where ***<authuser_password>*** is the password for the user assigned to the authuser parameter. The default is 'auracmgr', but if authuser has been set to a different user, that user's password will be required. See the section "Setting the CONSOLE_SERVERS Variable" on page 7-1 for information on setting up this account.

If security is being administered using `authfile`, the next lines entered will be:

`username:`***<authfile_user>***

`password:`***<authfile_user_password>***

The ***<authfile_user>*** specified will have the permissions assigned to them in the `authfile` file.

```
The viewer is now attached to hercules in read-only mode.
When you view a device in read-only mode, you cannot send
input to that system. To send input, you must attach in read-
write mode using an escape sequence. See "Escape Sequences"
on page 7-6.
```

- Attach in read-write mode to a managed device on the local server:

```
system# cmgr -a hercules
```

Security will be as described above, however, if authfile is used for authorization, if the user who logs in doesn't have attach permission, the device will be attached in read-only mode.

Similarly, if **-A** is used to force attach, if the user doesn't have attach and fattach permissions, any users already connected will not be disconnected and the device will be attached in read-only mode.

- View a device managed by a remote server:

  system# cmgr <*managed_device_name*>@<*remote_server_name*>
  If security is being administered using authuser, the next line entered will be:

  password:<*remote_server_authuser_password*>

  where ***<remote_server_authuser_password>*** is the password for the user assigned to the authuser parameter on the remote server. The default is 'auracmgr', but if authuser has been set to a different user, that user's password will be required. See the section "Setting the CONSOLE_SERVERS Variable" on page 7-1 for information on setting up this account.

  If security is being administered using authfile, the next lines entered will be:

  username:***<remote_server_authfile_user>***

  password:***<remote_server_authfile_user_password>***

The ***<remote_server_authfile_user>*** specified will have the permissions assigned to them in the authfile file.

The following commands have the described effects when **CONSOLE_SERVERS** is set to both the local and remote servers. When this is the case, cmgr treats devices on a remote server in the same way it treats local devices.

- List devices managed by the local server only:

```
system# cmgr -l
cmgr: must have system name
hercules apollo ulysses agamemnon
[cmgr viewer exiting]
```

- List devices managed by both local and remote servers:

```
system# cmgr
cmgr: must have system name
agamemnon@server1 apollo@server1 dagwood@server2
dilbert@server2 hercules@server1 lucy@server2
ulysses@server1
[cmgr viewer exiting]
```

- View a device managed by a remote server in read-only mode:

  system# cmgr <*managed_device_name*>
  If security is being administered using authuser, the next line entered will be:

  password:<*remote_server_authuser_password*>

where ***<remote_server_authuser_password>*** is the password for the user assigned to the `authuser` parameter on the remote server. The default is 'auracmgr', but if `authuser` has been set to a different user, that user's password will be required. See the section "Setting the CONSOLE_SERVERS Variable" on page 7-1 for information on setting up this account.

If security is being administered using `authfile`, the next lines entered will be:

```
username:<remote_server_authfile_user>
password:<remote_server_authfile_user_password>
```

The ***<remote_server_authfile_user>*** specified will have the permissions assigned to them in the `authfile` file.

- View either a local or remote device, specifying that output from the managed device to the terminal be 7 bits:

  ```
  system# cmgr -7 <managed_device_name>
  ```

  Use of this option may be necessary if all 8 bits are processed by the server, but are not tolerated by the terminal.

- View either a local or remote device, and specify a different escape character:

  ```
  system# cmgr -e % <managed_device_name>
  ```

  This causes all escape sequences to start with **%**.

- View a remote device, using encrypted communications to the server:

  ```
  system# cmgr -f -128 <managed_device_name>
  ```

Escape Sequences

If the `authfile` parameter is set instead of `authuser`, the users and permissions listed in the authfile file determine which escape sequences can be used by which user. See "authfile" on page 6-13 for further information. All escape sequences begin with the tilde character ("~"), unless it was changed using the `-e` option in the command line or the escape setting of the `AURACMGR_OPTIONS` environment variable. The available escape sequences are as follows:

~. (tilde period)	Terminate the session.
~CTRL/C	Terminate the session.
~CTRL/Z	Suspend the `cmgr` program. The session is resumed with `fg`.
~CTRL/L	Toggle local logging of the connection.
~a	Attach: While in read-only mode, attach (read-write mode) to the managed device. Requires attach permission.
~A	Force Attach: Force an attach (read-write mode) to the managed device. If someone else is attached (read-write), downgrade their connection to read-only. Requires attach and fattach permission.
~d	Detach from the managed device, i.e., make the connection read-only.
~q [VWTA?]	Query a server variable, as follows: V Version W who's connected to managed device T tail of the log file A all ? show options
~s [sT?]	Set a server variable as follows: s set terminal stty parameters (Requires stty permission.) T set tail length ? show options
~#	Send a BREAK (if currently attached.) The user is prompted to confirm this action, which is aborted if not confirmed. Requires break permission.
~?	Display help text on escape sequences

CARLO GAVAZZI
COMPUTING SOLUTIONS

CHAPTER 8

Warranty & Maintenance Information

Warranty on Hardware & Software

Aurora brand products Carlo Gavazzi Computing Solutions carry the following standard warranties:

Standard Hardware Warranty Policy

All Aurora brand hardware products are warranted against defects for two (2) years from the date of delivery. The Standard Warranty includes 90 days of free Technical Support, two (2) years product repair, and driver upgrades.

Standard Software Warranty Policy

Carlo Gavazzi Computing Solutions warrants that the physical media on which software is furnished will be free from defects in materials and workmanship, under normal use, for a period of (90) days from the date of shipment.

The Standard Warranty includes 90 days of Free Technical Support.

Make sure you complete the Warranty Registration form on page 8-2 and return it to Carlo Gavazzi Computing Solutions. Refer to Warranty information at www.gavazzi-computing.com for details on extended warranty plans.

Product Registration Form

Important! Please print, complete, and return this Product Registration Form to Carlo Gavazzi Computing Solutions' Customer Service and Support (CSS) Department at 508-588-0498. The information you provide here allows CSS to validate your warranty and inform you of software and hardware upgrades.

Purchase Order No.:_____**Sales Order No**.:_____
Serial No.:_____

Name/Title:_____

Company:_____

Street Address:_____

City:_____State:_____ Postal Code:_____

Country:_____

Phone:_____Fax:_____

Email Address:_____

Supplier Name:_____Date Purchased:_____

Supplier Address:_____

City:_____State:_____ Postal Code:_____

Country:_____

Supplier Phone:_____

Protocol/Software License Application

Product: ❑ X.25 ❑ HDLC ❑ Control Tower Version:_____

Workstation Type:_____ O/S Version:_____ Host ID:_____

Maximum Number of Ports:_____

Your Application

❑ Printer/Plotter Connectivity	❑ Internet Connectivity
❑Terminal/Instrumentation I/O	❑Telecom Service Provider
❑ Modem Pool	❑ Data Feed
❑ WAN Connectivity	❑ Other

Carlo Gavazzi Computing Solutions - 10 Mupac Drive Brockton, MA 02301 - USA
Phone: 508-588-6110 - Fax:508-588-0498 - E-mail: support@gavazzi-computing.com -
URL: www.gavazzi-computing.com

CARLO GAVAZZI
COMPUTING SOLUTIONS

Appendix A

Command Reference

User Commands

cmgr(1) **cmgr(1)**

NAME

cmgr - Aurora ControlTower Console Manager viewer program

SYNOPSIS

cmgr [-78aAlLPNv] [-d *debuglevel*] [-e c] [-f *keylength*] [-t *taillen*] [-o *options*] [*system[@server[:port]*

DESCRIPTION

cmgr establishes an interactive session with the console port of the named s*ystem*. If no system name is specified, a list of possible systems will be printed.

If **@server** follows the system name, a TCP/IP connection will be made to the named server, and the system name is used literally (no abbreviations accepted). The server name can be followed by a colon and a port number (or name from the **services**(4) file) to contact on the remote server. If the CONSOLE_SERVERS environment variable contains a comma seperated list of servers (and the -l option is not

given), the servers will be contacted in turn to retrieve the list of all possible system names. Each server name can optionally be followed by a colon and a port number (or name from the **services**(4) file) to use to contact the remote server. If no system name is present on the command line, all the system names (and the server to which they are attached) will be sorted and printed. If a system name is present, it may be an unambiguous prefix. If the prefix is ambiguous, all matching system names will be printed.

If the **CONSOLE_SERVERS** environment variable is not set (or the **-l** option is given), the connection will be made locally, using Unix- domain sockets. This requires file access permission to the directory in which the sockets are located, and will not require a password to establish a connection. If no system name is present on the command line, all the system names will be sorted and printed. If a system name is present, it may be an unambiguous prefix. If the prefix is ambiguous, all matching system names will be printed.

OPTIONS

-7 Output only 7 bits of data to the terminal. This may-be necessary if all 8 bits are being processed by the server, but are not tolerated by the user as terminal.

-8 Neutralizes the effect of the -7 option.

-a Attach to the system console as soon as the connection is established. By default sessions are view-only, and an escape sequence attach command (see below) must be typed to send input to the remote console port. If someone is already attached, a view-only connection will be established.

-A Force an attach to the system console as soon as the connection is established. If someone is already attached, their connection will be reduced to view-only.

-v Neutralizes the effect of the -a and -A options.

-d *debuglevel* Set program debug level.

-e *escapechar* Set escape character. If escapechar is a single char-

acter it is used directly as the escape character. If escapechar is a multi-character sequence starting with a digit, it is interpreted according to strtol(3). If escapechar is none there is no escape character. Default escape character is tilde (~).

-f *keylength* Client Encryption select parameter, where ***keylength*** is 0 or 128. -f 128 selects 128-bit encryption, -f 0 disables encryption. Servers can be set to restrict TCP/IP con- sections to minimum key lengths through the ForceEncrypt=true configuration directive.

-l Force a connection to be made locally, even if the **CONSOLE_SERVERS** environment variable is set.

-L Neutralizes the effect of the **-l** option.

-p *port* changes the default TCP/IP port or service name used to contact remote location brokers.

-P On TCP/IP connections, use location broker pass-thru feature. This only works with version 2.00 or later remote systems. When using this option, the only TCP connections made will be on the auracmgr/tcp port (364) which facilitates use across firewalls, Network Address Translation (NAT), and ssh, stunnel, or other port of warding. The network connection is actually passed from the locbrok(8) process to the **conserv(8)** process, so there is no performance penalty.

-N Neutralizes the effect of the -P option.

-v Connect in view-only mode (neutralizes -a and -A).

-o *options* Set AURACMGR_OPTIONS style options.

Escape Sequences

Lines that you type which start with the tilde character are \(lqescap sequences\(rq (the escape character can be changed using the -e option, see above).

~. Terminate the session.

~CTRL/C	Terminate the session.
~CTRL/Z	Suspend the cmgr program.
~CTRL/L	Toggle local logging of the connection.
~a	Attempt to attach (read-write) to the console.
~A	Force an attach (read-write) to the console; if anyone is currently attached, their connection will be downgraded to view only.
~d	Detach from the remote console (make the connection view-only).
~q	Query server; a single character specifies the information return: A (show All) W (show Who is connected to server), (show Tail of log file) V (show Versions of server and console programs)? (show available options).
~s	Set a server variable (only if currently attached) a single character specifies the information to change; s (set stty parameters, a subset of the stty(1) command, options include: **crtscts -crtscts cstopb -cstopb parenb -parenb parext -parex parodd -parodd ixon -ixon ixoff -ixoff istrip -istrip cs5 cs6 cs7 cs8** or a baud rate), T (set distance back tail query will display in log file). ? (show available options).
~#	Send a BREAK (only if currently attached). The user is prompted to confirm this action, which will be aborted if not confirmed.
~?	Display help on escape sequences.

All other characters typed are sent to the remote system when attached. If not attached, the bell is rung for each character typed.

ENVIRONMENT

CONSOLE_SERVERS

see above.

AURACMGR_OPTIONS

AURACMGR_OPTIONS Establish per-user defaults before checking command line options. AURACMGR_OPTIONS consists of a sequence of strings (and values) seperated by commas, one or more of;

attach	see **-a** option.
Attach	see **-A** option.
7bit	see **-7** option.
debug=number	see **-d** option.
escape=string	see **-e** option.
tail=number	see **-t** option.
port=string	see **-p** option.
local	see **-l** option.
nolocal	see **-L** option.
passthru	see **-P** option.
nopassthru	see **-N** option.

SEE ALSO

services(5), locbrok(8)

13 May 2001 **cmgr(1)**

File Formats

config(4)

NAME

config – ControlTower Console Manager server configuration file

SYNOPSIS

```
/etc/AURAcmgr/DEFAULT

/etc/AURAcmgr/LOCAL

/etc/AURAcmgr/group/group.grp

/etc/AURAcmgr/[group/]system.cfg
```

DESCRIPTION

The ControlTower Console Manager **conserv** and **logcheck** programs read **DEFAULT**, site **LOCAL**, group, and per-system configuration files of the format described here. The **DEFAULT** file contains all global default values, and should not be edited. The **LOCAL** file is then read to allow avoid losing local changes to the **DEFAULT** file that might be lost in an upgrade. If the system config file is located in a subdirectory, that directory must contain a group configuration file with the same name as the directory, and the suffix
.cfg"."

Finally, **the** *system***.cfg** file is read to supply values unique to a single system. The name of the configuration file determines the name of the managed system as known to the ControTower console management server, and need not be the official name of the server.

File Format

Lines which start with a **#** character are treated as comments and ignored. Configuration lines are of the form *parameter***=***value*, where *parameter* is a case-insensitive parameter name, and *value* is the parameter value.

Parameter Syntax

Each parameter takes a value with one of the following syntaxes;

int	Any integer value. A prefix of **0x** means value will be interpreted as base 16 (hex). A prefix of **0** means value will be interpreted as base 8 (octal). Otherwise the value is interpreted as base 10 (decimal).
boolean	A boolean value, one of: **1/t/true/y/yes** to enable a parameter, or on of **0/f/false/n/no** to disable a parameter.
string	An arbitrary string.
mode	File protection "mode", ether an octal constant (no

	leading digit required), or a symbolic value *[***ogua***]=[***rwxs***]+{,...}* (see **chmod**(1) man page).
uid	User id: decimal value or a user name from the **passwd**(4) file.
gid	Group id: decimal value or group name from the **group**(4) file.
stty	sequence of tokens/values (see **stty**(1) man page). Character size; **cs5**, **cs6**, **cs7**, **cs8**. Line speed (supported speeds depend on underlying hardware and operating system. Custom speeds are not supported); **50**, **75**, **110**, **150**, **200**, **300**, **600**, **1200**, **2400**, **4800**, **9600**, **19200**, **38400**, **57600**, **76800**, **115200**, **153600**, **230400**, **307200**, **460800**. Flags (may be prefixed with '**–**' to disable); **crtscts**, **cstopb**, **parenb**, **parext**, **parodd**, **ixon**, **ixoff**, **istrip**.

Serial line parameters

device	Syntax: *string*. **device** is the only parameter which must appear in the *system*.cfg file. This is the path of a tty device for the managed system. Call out devices **/dev/cua*** are typically used, to ignore changes in the state of the Data Carrier Detect (DCD) control line.
stty	Syntax: *stty.* Sets the initial terminal modes for the managed system serial connection. Any parameters missing from both the system and DEFAULT configuration will be left unmodified from system defaults (admintool terminal settings have no effect).
ttychanges	Syntax: *boolean*. Allow attached clients to change serial line parameters.
uulock	Syntax: *boolean*. Honor and create uucp compatible lock files for the serial port.
exclusive	Syntax: *boolean*. Set operating system 'exclusive access' flag on the serial port using the **TIOCEXCL** ioctl. Prevents non-superuser processes from opening the serial port.

breakstring Syntax: *string*. String to send to managed system instead of BREAK signaling. The following escape sequences are allowed: **\n** (newline), **\r** (return), **\t** (tab), *****OOO* (octal value), **\x***XX* (hex value).

Log file parameters

logdir Syntax: *string*. Specify the absolute path to the directory for per-system log files. Log file names are by default the *system* name of the managed system, but can be explicitly specified using the **logfile** parameter. If the managed device is a member of a group, the device log file will be created in a subdirectory having the same name as the group.

logfile Syntax: *string*. Specify the absolute path to the file to which log output will be written. Defaults to `logdir/<servername>`, but can be customized to a pathname for each server individually or all servers combined.

logfilter Syntax: *string*. The path to an optional log file filter file. The path may be relative. Each line in the **logfilter** file starts with a filter type, one of **keep**, **drop**, or **alert**, followed by a delimited POSIX 1003.2 extended regular expression (see **regex**(7)), which may be followed by optional "tag" text. The delimiter character used to bracket the regular expression must not appear within the regular expression.

loginput Syntax: *boolean*. log user input (from attached user(s)). Will cause "double echo" of user input. No-echo input (passwords) will be logged!

loglinestamp Syntax: *string*. Format string passed to **strftime**(3) to format timestamp on each line written to log file. If empty, lines are not timestamped.

logmessages Syntax: *boolean*. Log connect/disconnect/force messages (normally sent to connected users) to managed system log file (including parameter set messages).

logmode Syntax: *mode*. Protection for per-system log file.

May be empty. Ignored if mode=0. Logfile mode is set on each (re)open for append.

logowner Syntax: *uid*. Owner for per-system log file. May be empty. The Log file owner is set on each (re)open for append.

loggroup Syntax: *gid*. Group for per-system log file. May be empty. Log file group is set on each (re)open for append.

logstamp Syntax: *integer*. Determines how often to timestamp the log file in minutes; one of: **10**, **20**, **30**, **60** or zero to disable periodic timestamps.

logstampformat Syntax: *string*. Format string passed to **strftime**(3) to format periodic logfile timestamps. If empty, periodic timestamps are not output.

logcompress Syntax: *string*. Pathname of a program for **logcheck**(8) to use to compress old log files.

Local connection control parameters

localenable Syntax: *boolean*. True to allow local (unix domain) socket connections. This parameter was called **unixenable** in version 1.00 (which is still accepted as an alias).

localauth Syntax: *boolean*. True to force local (unix domain) socket connection users to be prompted for password (see **authuser** below). This parameter was called **unixauth** in version 1.00. (which is still accepted as an alias).

Network access control parameters

tcpenable Syntax: *boolean*. True to allow TCP/IP connections.

tcpallow Syntax: *string*. If non-null, TCP connections will only be accepted if the remote host matches a member of this list of comma seperated TCP hosts or networks to allow connections from. Hosts may be host names or IP addresses. Each may be followed with a forward slash (/) and an optional mask in dotted oc-

tet format, hex, or decimal network mask length. All of the following have the same effect: **/255.255.255.0**, **/0xffffff00**, **/24**. The mask determines which bits in the IP addresses will be examined: Any bit position with a zero mask bit will be ignored.

tcpreject Syntax *string*. If non null, TCP connections will be rejected if the remote host matches a member of this list (see **tcpallow** for syntax).

authuser Syntax: *string*. The name of a local user remote users must supply the password for when connecting. If not set (empty), or the user does not exist, no one can connect over the network (or locally if **unixauth** (see above) is set).

authfile Syntax: *string*. The path to an optional per-user authorization file, which contains a list of authorized users and their capabilities. The path may be relative. If the **authfile** parameter is not specified, all users must authenticate as the user specified by the **authuser** parameter. The authfile format is: *username* followed by one or more of the following; **connect** (may connect to server), **attach** (may attach in r/w mode), **fattach** (may force others off), **stty** (may change tty params), **break** (may send break), **all** (all of the above) seperated by plus (**+**) signs to add capabilities or minus (**-**) signs to subtract them.

Idle time limit parameters

detachidle Syntax: *integer*. If non-zero, the maximum time in minutes before detaching (demoting to view-only) an idle attached viewer. If zero, no idle limit is enforced.

disconnectidle Syntax: *integer*. If non-zero, the maximum time in minutes before disconnecting an idle viewer regardless of whether viewer is attached or view-only. If zero, no idle limit is enforced.

FILES

`/etc/AURAcmgr/DEFAULT`	default values
`/etc/AURAcmgr/LOCAL`	site local default values
`/etc/AURAcmgr/group/group.grp`	group default values
`/etc/AURAcmgr/]system.cfg`	per-system configuration

SEE ALSO

conserv(8), **logcheck**(8).

config(5) config(5)

NAME

config - Aurora ControlTower Console Manager server configuration file

SYNOPSIS

/etc/AURAcmgr/DEFAULT

/etc/AURAcmgr/LOCAL

/etc/AURAcmgr/group/group.grp

/etc/AURAcmgr/[group/]device.cfg

DESCRIPTION

The Aurora ControlTower Console Manager conserv and logcheck programs read DEFAULT, site LOCAL, group, and per-device configuration files of the format described here. The DEFAULT file contains all global default values, and should not be edited. The LOCAL file is then read to allow avoid losing local changes to the DEFAULT file that might be lost in an upgrade. If the device config file is located in a subdirectory, that directory must contain a group configuration file with the same name as the directory, and the suffix .cfg. Finally, the device.cfg file is read to supply values unique to a single device. The name of the configuration file determines the name of the managed device as known to the ControTower console management server, and need not be the official name of the server.

FILE FORMAT

Lines which start with a # character are treated as comments and ignored. Configuration lines are of the form parameter=value, where parameter is a case-insensitive parameter name, and value is the parameter value.

PARAMETER SYNTAX

Each parameter takes a value with one of the following syntaxes:

int Any integer value. A prefix of 0x means value will be interpreted as base 16 (hex). A prefix of 0 means value will be interpreted as base 8 (octal). Otherwise the value is interpreted as base 10 (decimal).

boolean A boolean value, one of: 1/t/true/y/yes to enable a parameter or on of 0/f/false/n/no to disable a parameter. string An arbitrary string. The magic string *novella* resets the parameter so that it appears no value has ever been set. This is useful for riding overriding defaults set in a highest level configuration for optional parameters.

mode File protection “mode”, ether an octal constant (no leading digit required), or a symbolic value [ogua]=[rwxs]+{,...} (see chmod(1) man page).

uid User id: decimal value or a user name from the passwd(5) file.

gid Group id: decimal value or group name from the group(5) file.

stty sequence of tokens/values (see stty(1) man page). Character size; cs5, cs6, cs7, cs8. Line speed (supported speeds depend on underlying hardware and operating system. custom speeds are not supported); 50, 75, 110, 150, 200, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200, 153600, 230400, 307200, 460800. Flags (may be prefixed with â-â to disable); crtscts, cstopb, parenb,

parext, parodd, ixon, ixoff, istrip.

Serial line parameters

device Syntax: string. device is the only parameter which must appear in the device.cfg file. This is the path of a tty device for the managed device. Call out devices /dev/cua* are typically used, to ignore changes in the state of the Data Carrier Detect (DCD) control line.

stty Syntax: stty. Sets the initial terminal modes for the managed device serial connection.

ttychanges Syntax: boolean. Allow attached clients to change serial line parameters.

uulock Syntax: boolean. Honor and create uucp compatible lockfiles for the serial port.

exclusiv Syntax: boolean. Set operating system "exclusive access" flag on the serial port using the TIOCEXCL ioctl. Prevents non-superuser processes from opening the serial port.

breakstring Syntax: string. String to send to managed device instead of BREAK signaling. The following escape sequences are allowed: \n (newline), \r (return), \t (tab), \OOO (octal value), \xXX (hex value).

Log file parameters

logdir Syntax: string. Specify the absolute directory for per-device log file. Log files are always the device name of the manage device. logfilter Syntax: string. The path to an optional log file filter file. If the path name is not absolute (does not begin with a slash), it will be interpreted as relative to the directory in which the device .cfg file was found. Lines starting with a # character are discarded as comments. Each line in the logfilter file starts with a filter type: drop or keep, followed by a delimited POSIX 1003.2 extended regular expression (see regex(7)), The delimiter character used to bracket the regular expression

must not appear within the regular expression.

loginput Syntax: boolean. log user input (from attached user(s)). No echo input (including passwords) will be logged!

loglinestamp Syntax: string. Format string passed to strftime(3) to format timestamp on each line written to log file. If empty, lines are not timestamped.

logmessages Syntax: boolean. log connect/disconnect/force messages (normally sent to connected users) to managed device log file (including parameter set messages).

logmode Syntax: mode. Protection for per-device log file. May be empty. Ignored if mode=0. Logfile mode is set on each (re)open for append.

logowner Syntax: uid. Owner for per-device log file. May be empty. The Log file owner is set on each (re)open for append.

loggroup Syntax: gid. Group for per-device log file. May be empty. Log file group is set on each (re)open for append.

logstamp Syntax: integer. Determines how often to timestamp the log file in minutes; one of: 10, 20, 30, 60 or zero to disable periodic timestamps.

logstampformat Syntax: string. Format string passed to strftime(3) to format periodic logfile timestamps. If empty, periodic timestamps are not output.

Local connection control parameters

localenable Syntax: boolean. True to allow local (unix domain) socket connections. NOTE: This parameter was called unixenable in version 1.00 (which is still accepted as an alias).

localauth Syntax: boolean. True to force local (unix domain) socket connection users authenticate (see authuser and authfile below). If false, local (unix domain) socket connections will not be prompted for a user-

name or password. This is safe so long as the /var/lib/AURAcmgr/sockets/sock is protected to only allow access to authorized users. NOTE: This parameter was called unixauth in version 1.00. (which is still accepted as an alias).

Network access control parameters

tcpenable Syntax: boolean. True to allow TCP/IP connections.

tcpallow Syntax: string. If non-null, TCP connections will only be accepted if the remote host matches a member of this list of comma seperated TCP hosts or networks to allow connections from. Hosts may be host names or IP addresses. Each may be followed with a forward slash (/) and an optional mask in dotted octet format, hex, or decimal network mask length. All of the following have the same effect: / 255.255.255.0, /0xffffff00, /24. The mask determines which bits in the IP addresses will be examined: Any bit position with a zero mask bit will be ignored.

tcpreject Syntax string. If non null, TCP connections will be rejected if the remote host matches a member of this list (see tcpallow for syntax).

authuser Syntax: string. If authfile is not set, this is the name of a local user remote users must supply the password for when connecting. If not set (empty), or the user does not exist, no one can connect over the network (or locally if unixauth (see above) is set). If authfile is set, authuser only applies to version 1 viewers.

authfile Syntax: string. The path to an optional per-user authorization file, which contains a list of authorized users and their capabilities. If the path name is not absolute (does not begin with a slash), it will be interpreted as relative to the directory inwhich the device .cfg file was found. If the authfile parameters not specified, all users must authenticate as the user specified by the authuser parameter. The authfile format is: user- name followed by one or more of the

following; connect (may connect to server), attach (may attach in r/w mode), fattach (may force others off), stty (may change tty params), break (may send-break), all (all of the above) seperated by plus (+) signs to add capabilities or minus (-) signs to subtract them.

Idle time limit parameter

detachidle Syntax: integer. If non-zero, the maximum time in minutes before detaching (demoting to view-only) an idle attached viewer. If zero, no idle limit is enforced.

disconnectidle Syntax: integer. If non-zero, the maximum time in minutes before disconnecting an idle viewer regardless of whether viewer is attached or view-only. If zero, no idle limit is enforced.

Misc. parameters

attires Syntax: string. Can be set to one of three values: “vt100 DC1”,“vt100 DC2” or “vt100 DS”. If set, the server will send adequate response to the vt100 queries, even when no client is connected. This can be used to insure that some PC console are active since they rely on the vt100 sequence during boot to determine if they are connected and, if so, the baud rate to use.

FILES

/etc/AURAcmgr/DEFAULT	default values
/etc/AURAcmgr/LOCAL	site local default values
/etc/AURAcmgr/group/group.grp	group default values
/etc/AURAcmgr/[group/]device.cfg	per-device configuration

SEE ALSO

is not specified, all users must authenticate as the user specified by the authuser parameter. The authfile format is: username followed by one or more of the following; connect (may connect to server), attach (may attach in r/w mode), fattach (may

force others off), stty (may change tty params), break (may sendbreak), all (all of the above) seperated by plus (+) signs to add capabilities or minus (-) signs to subtract them.

Idle time limit parameters

detachidle Syntax: integer. If non-zero, the maximum time in minutes before detaching (demoting to view-only) an idle attached viewer. If zero, no idle limit is enforced.

disconnectidle Syntax: integer. If non-zero, the maximum time in minute before disconnecting an idle viewer regardless of whether viewer is attached or view-only. If zero, no idle limit is enforced.

Misc. parameters

attires Syntax: string. Can be set to one of three values: “vt100 DC1”, “vt100 DC2” or “vt100 DS”. If set, the server will send adequate response to the vt100 queries, even when no client is connected. This can be used to insure that some PC console are active since they rely on the vt100 sequence during boot to determine if they are connected and, if so, the obdurate to use.

FILES

/etc/AURAcmgr/DEFAULT	default values
/etc/AURAcmgr/LOCAL	site local default values
/etc/AURAcmgr/group/group.grp	group default values
/etc/AURAcmgr/[group/]device.cfg	per-device configuration

SEE ALSO

conserv(8), logcheck(8).

26 May 2001 **config(5)**

cmgrd(8) cmgrd(8)

NAME

cmgrd - Aurora ControlTower Console Manager server start/stop script

SYNOPSIS

/etc/init.d/cmgrd <command> [system group range]

DESCRIPTION

The cmgrd script is used by the system to start and stop the ControlTower system. See the documentation of your distribution on how to enable services during boot.

The cmgrd script must have at least a command argument. Following the command, there can be other arguments to specify which services are to be affected by the command.

COMMAND: start

If no other argument is given start first checks if locbrok(8)) is run king. If it is not, it will start it. It then starts a conserv(8) process for each file ending in .cfg in the /etc/AURAcmgr directory. If other arguments are given, start launches a locbrok(8) process if needed, and a conserv(8) for each service named in the argument list. If the argument specifies a group directory, a conserv(8) will be launched for service defined in the group directory . if the argument specifies a range, a conserv(8) will be launched for service in the range.

COMMAND: stop

If no other argument is given stop send the TERM signal to all the conserv processes running on the system. It then sends a TERM signal to the **locbrok(8))** process.

If other arguments are given, stop sends a TERM signal to the **conserv(8)** process corresponding to each service named in the argument list. If the argument specifies a group directory, a **conserv(8)** for each service in that group is terminated. If other arguments are given, stop sends a TERM signal to the **conserv(8)** process corresponding to each service named in the argument list. If the argument specifies a group directory, a **conserv(8**) for each service in that group is terminated. if the argument specifies range, the **conserv(8)** process is terminated for all the services in the range.

COMMAND: status

Reports whether the **locbrok (8)** process was started or not. Other arguments are ignored.

SERVICE NAME ARGUMENT

The start and stop commands can be given a list of service name arguments. Here is how those parameters can be specified.

The argument can be a service name. The definition of that service is contained in the file <service>.cfg under /etc/AURAcmgr.

Note that the service may include a group name. E.g., Groups1/Service0

The argument can be a group name. In that case, all the services defined in that group are affected.

The third type of service name argument is the range. This type can only be used when one uses the following naming convention for the services: All the service names should consist in a prefix followed by a number of digits not starting with 0. (e.g. port1, port21,...). The range can then be defined by using an optional group name, followed by the prefix, followed by the start of the range, followed by a dash (-) followed by the end of the range. For example: Group1/port5-12

FILES

/std./AURAcmgr/*.cfg	config files

SEE ALSO

cmgrd-config(5), conserv(8), locbrok(8),

12 September 2005 **cmgrd(8)**

filtertest(8) **filtertest(8)**

NAME

filtertest - Aurora ControlTower Console Manager log filter test program

SYNOPSIS

filtertest [-n] [-q] filterfile [inputfile]

DESCRIPTION

filtertest reads an Aurora ControlTower Console Manager log filter file, checks the file for syntax, and reads an input file and applies the filters to each line of the input file. If no input file is specified, lines are read from the standard input stream. Each match is reported on the standard output, and a summary of each type of match (keep, drop, alert) is reported on standard errors when end of file is reached on standard input.

When the -n (no filter) option is specified, filtertest will exit with zero (true) status after successfully parsing the filter file. When the -q filter is specified, matches are not reported on standard output. A summary is still reported on standard error.

SEE ALSO

conserv(8).

29 Oct. 2000 **filtertest(8)**

Maintenance Procedures

conserv(8) **conserv(8)**

NAME

conserv - Aurora ControlTower Console Manager server process

SYNOPSIS

conserv [**-d** *debuglevel*] [**-o** *parameter*=*value*] system

DESCRIPTION

Aurora ControlTower Console Manager server launches a conserv for each managed system. conserv reads the system.cfg file (see **config(5))** and opens the serial port specified by the device parameter. conserv logs all managed system output in a file named system in the directory specified by the logdir configuration parameter. Users can connect to the conserv process using the **cmgr(1)** program. con**serv(8)** is normally launched during the normal boot process by the /etc/init.d/cmgrd script, but can be started by hand for debugging. Any number of options may be given, each with a parameter=value pair to override values in the system.cfg file. The -d option can be used to specify a debug level, which if non-zero keeps conserv from detaching from the terminal so that debug messages can be seen. Increasing debug levels increase the amount of debug output.

FILES

/etc/AURAcmgr/system.cfg	configuration file
/var/run/system	process id file
/var/lib/AURAcmgr/sockets/system	unix-domain socket endpoint

SEE ALSO

cmgr-config(5), conserv(8), locbrok(8),

16 Jan 2000 conserv(8)

convert(8)

NAME

convert – ControlTower Console Manager config file conversion tool

SYNOPSIS

convert [**-f**] [**-o** *outputdir*] [*inputfile...*]

DESCRIPTION

convert reads input files (or the standard input if none are specified) that are tab or colon delimited and creates ControlTower Console Manager config files. The first column of the input is the managed system name, and the second is the serial device to which the managed system console is attached. Any remaining information is

discarded. If a configuration file already exists, the entry will be skipped, unless the **-f** option is used, in which case the existing file will be saved as a.bak file. The **-o** option specifies the configuration file output directory.

FILES

`/opt/AURAcmgr/config/system.cfg`	system configuration files

SEE ALSO

config(4), **conserv**(8).

Filtertest(8)

NAME

filtertest – ControlTower Console Manager log filter test program

SYNOPSIS

filtertest [**-n**] [**-q**] *filterfile* [*inputfile*]

DESCRIPTION

filtertest reads a ControlTower Console Manager log filter file, checks the file for syntax, and reads an input file and applies the filters to each line of the input file. If no input file is specified, lines are read from the standard input stream. Each match is reported on the standard output, and a summary of each type of match (keep, drop, alert) is reported on standard errors when end of file is reached on standard input.

When the **-n** (no filter) option is specified, *filtertest* will exit with zero (true) status after successfully parsing the filter file.

When the **-q** filter is specified, matches are not reported on standard output. A summary is still reported on standard error.

SEE ALSO

conserv(8).

locbrok(8)

NAME

locbrok – ControlTower Console Manager server Location Broker

SYNOPSIS

locbrok [**-d** *debuglevel*]

DESCRIPTION

The ControlTower Console Manager Location Broker reads and enforces the terms of the product licence file, and keeps a database of managed system names and the **TCP/IP** port the **conserv**(8) process for that managed system is available at. When a **cmgr**(1) is run remotely it first contacts one or more Location Brokers (on one or more servers) in order to find out what managed systems are available, what server they are attached to, and on which **TCP** port the **conserv (8)** process can be reached.

The **locbrok** process is normally launched by the **start**(8) script, and killed by the **stop**(8) script.

FILES

`/etc/AURAcmgr/license.dat`	license file
`/etc/AURAcmgr/pids/.locbrok`	process id file
`/etc/AURAcmgr/sock/.system/locbrok2`	unix-domain socket endpoint

SEE ALSO

cmgr(1), **conserv**(8), **start**(8), **stop**(8).

filtertest(8) filtertest(8)

NAME

filtertest - Aurora ControlTower Console Manager log filter test program

SYNOPSIS

filtertest [**-n**] [**-q**] filterfile [*inputfile*]

DESCRIPTION

filtertest reads an Aurora ControlTower Console Manager log filter file, checks the file for syntax, and reads an input file and applied the filters to each line of the input file. If no input file is specie fed, lines are read from the standard input stream. Each

match is reported on the standard output, and a summary of each type of match (keep, drop, alert) is reported on standard errors when end of file is reached on standard input.

When the -n (no filter) option is specified, filtertest will exit with zero (true) status after successfully parsing the filter file. When the -q filter is specified, matches are not reported on standard output. A summary is still reported on standard error.

SEE ALSO

conserv(8).

29 Oct. 2000 filtertest(8)

locbrok(8) locbrok(8)

NAME

locbrok - Aurora ControlTower Console Manager server Location Broker

SYNOPSIS

locbrok [-d debuglevel]

DESCRIPTION

The Aurora ControlTower Console Manager Location Broker reads and enforces the terms of the product licence file, and keeps a database of managed system names and the TCP/IP port the conserv(8) process for that managed system is available at. When a cmgr(1) is run remotely it first contacts one or more Location Brokers (on one or more servers) in order to find out what managed systems are available, what server they are attached to, and on which TCP port the conserv (8) process can be reached.

The locbrok process is normally launched by the /etc/init.d/cmgrd start command, and killed by the /etc/init.d/cmgrdstop(8) command.

FILES

/etc/AURAcmgr/license.dat	license file
/var/run/.locbrok	process id file
/var/lib/AURAcmgr/sockets/.system/locbrok2	unix-domain socket endpoint

SEE ALSO

cmgr(1), conserv(8),

16 Jan 2000 locbrok(8)

CARLO GAVAZZI
COMPUTING SOLUTIONS

Appendix B

Default Configuration File

Introduction

The DEFAULT configuration file (`/opt/AURAcmgr/config/DEFAULT`) shown here specifies the default configuration for devices managed by a ControlTower server. These configuration specifications apply to every managed device unless overriden in the LOCAL configuration file, the <*group_name*>/<*group_name*>.grp file, or the configuration file for that device (`/etc/AURAcmgr/`<*device_name*>.cfg.)

Default Configuration File Example

```
# Aurora ControlTower Console Manager DEFAULT configuration
#
# ******************** DO NOT EDIT THIS FILE ********************
#
# This file is read before the LOCAL file, <group>/<group>.grp files,
# and <system>.cfg files. Any changes to these defaults should be
```

```
# made by adding lines to the LOCAL, or per-group configuration files.
#
######################################################################
# COPYRIGHT (c) 1998, 2005 BY CARLO GAVAZZI COMPUTING SOLU-
TIONS, INC.
# BROCKTON, MA.
#
# THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE
USED AND
# COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE
AND WITH
# THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFT-
WARE OR
# ANY OTHER COPIES THEREOF MAY NOT BE PROVIDED OR OTHER-
WISE MADE
# AVAILABLE TO ANY OTHER PERSON. NO TITLE TO AND OWNERSHIP
OF THE
# PROGRAM IS HEREBY TRANSFERRED.
#
# THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE
WITHOUT
# NOTICE AND SHOULD NOT BE CONSIDERED AS A COMMITMENT BY
CARLO
# GAVAZZI COMPUTING SOLUTIONS, INC.
#
#
######################################################################
#              **** NOTE WELL ****        #
######################################################################
#
```

```
# With very few exceptions, a value for each parameter MUST be
# specified. NO default values are present in the code, so
# values must be specified here, or in a per-system config file.
# All parameters for which a default can be reasonably picked
# appear here.
#
#####################################################################
# serial line parameters
# serial line parameters
#
# devicename for serial port attached to system console
# syntax: string
#device=
#
# set O_EXCL “exclusive open” bit on serial port open
# syntax: boolean
exclusive=true
#
# create (and honor) UUCP-compatible lock files for the serial port
# syntax: boolean
uulock=true
#
# tty mode.
# syntax: one or more tokens separated by commas or spaces
# tokens;
#     integer (speed/baud)
#     cs5 cs6 cs7 cs8
```

```
#     flag
#     -flag
# flags:
#     crtscts cstopb parenb parodd ixoff ixon istrip
#
# ALL flags/parameters should appear here in DEFAULT file. subsequent
# stty configuration (in <system>.cfg or with -o on command line, or
# via console program "set" command) change ONLY the bits which are
# specified (all other remain the same).
stty=9600 cs8 -crtscts -cstopb -parenb -parodd -ixoff -ixon istrip
#
# allow client programs to change serial line parameters
# syntax: boolean
ttychanges=true
#
# string to send instead of BREAK signal (optional)
# syntax: string
# the following escape sequences are allowed;
#     \r \n \t \ooo (1 or more octal digits) \xXX (two hex digits)
#     \r \n \t \ooo (1 or more octal digits) \xXX (two hex digits)
#breakstring=
#
######################################################################
#
# Logfile parameters
# All system console output is saved in a logfile.
#
# directory for all log files (must be absolute)
# syntax: string (path)
```

```
logdir=/var/log/AURAcmgr
#
# log client (user) input in logfile (THIS INCLUDES PASSWORDS!!)
# all output (including echo) is always saved in the logfile
# syntax: boolean
loginput=false
#
# log messages sent to users (user connect/disconnects) in logfile
# (serial line change and break messages are always logged)
# syntax: boolean
logmessages=true
#
# owner for log files
# syntax: user name or uid
logowner=root
#
# group for log files
# syntax: group name or gid
loggroup=root
#
# mode for log files
# syntax: octal mode (e.g.; 0600) or comma separated sequence
#     of symbolic absolute modes strings [uoga]=[rwxs]+
logmode=u=rw
#
# Optional: strftime(3) format used to timestamp lines in logfile
#     if empty, lines are not time-stamped
```

```
# syntax: string
# syntax: string
loglinestamp=%c
#
# Optional: how often to timestamp logfile in minutes; one of: 10, 20, 30, 60
#     or zero to disable.
# syntax: integer
logstamp=60
#
# strftime(3) format used to output periodic logfile timestamps;
#     if empty, no periodic timestamps will be output
# syntax: string
logstampformat=********** %c **********
#
######################################################################
#
# Authorization parameters
#
# The name of the user remote users must supply the password for.
# If authfile (below) is specified, authuser only applies to version 1
# viewers.
# syntax: string (user name)
authuser=auracmgr
#
# The path to an optional per-user authorization file which contains a
# list of authorized users and their capabilities. The path may be
# relative (to group or config directory). If not specified, all
# users must authenticate as “authuser”, if set users will be prompted
# for a user name.
```

```
#
# To override a value set in a higher level configuration file, use the
# magic string *novalue*
#
# syntax: string (path)
authfile=/etc/AURAcmgr/authusers
#
#####################################################################
#
# Idle time parameters
# Idle time parameters
#
# If non-zero, the maximum time in minutes before disconnecting an
# idle viewer regardless of whether view is attached/view-only.  If
# zero, no idle limit is enforced.
# syntax: integer (minutes)
disconnectidle=0
#
# If non-zero, the maximum time in minutes before detaching (demoting
# to view-only) an idle attached viewer.  If zero, no idle limit is
# enforced.
# syntax: integer (minutes)
detachidle=0
#
#####################################################################
# local (Unix domain) socket parameters
#
# Allow local (Unix-domain) connections
```

```
# was called unixenable in Version 1.00
# (old name still accepted)
# syntax: boolean
localenable=true
#
# Require password on local (Unix-domain) connections;
# was called unixenable in Version 1.00
# (old name still accepted)
# syntax: boolean
localauth=false
#
######################################################################
# TCP socket parameters
#
# allow TCP/IP connections
# syntax: boolean
tcpenable=true
#
# The following have no default value, and may be left blank.
# syntax: comma separated list of host/mask pairs.
# syntax: comma separated list of host/mask pairs.
#
# The host may be a name (from /etc/hosts or DNS) or dotted decimal octets
# (nnn.nnn.nnn.nnn).  The mask is optional, and can be used to specify
# which bits of the host address are to be examined.
#
# 1) a single decimal number (/24) signifying the number of high-order
#    bits set in the mask
# 2) four dotted decimal octets (/255.255.255.0)
```

```
# 3) a single hexadecimal value (/0xffffff00)
#
# if no mask is supplied, all bits in the host address are examined,
# so "host/32" is the same as "host"
#
# If set a host must match an entry in "tcpallow" in order to be accepted.
# To override a value set in a higher level configuration file, use the
# magic string *novalue*
# syntax: string (acl)
#tcpallow=
#
# If set a host must NOT match an entry in "tcpdeny" in order to be accepted
# To override a value set in a higher level configuration file, use the
# magic string *novalue*
# syntax: string (acl)
#tcpdeny=
#
#####################################################################
#            AUTOBAUD function
#
#  Some PC (mostly Dell apparently) have a serial console that
#  uses some VT100 control parameters to figure out if the
# console is connected, and if so what betrayer to use.
#
# The problem with those consoles is that if no client is connected
# to the line when the host is booted, the serial console does not
# get any answer to its vt100 requests and the console is not
```

```
# becoming active.
#
#
# if no mask is supplied, all bits in the host address are examined,
# so “host/32” is the same as “host”
#
# If set a host must match an entry in “tcpallow” in order to be accepted.
# To override a value set in a higher level configuration file, use the
# magic string *novalue*
# syntax: string (acl)
#tcpallow=
#
# If set a host must NOT match an entry in “tcpdeny” in order to be accepted
# To override a value set in a higher level configuration file, use the
# magic string *novalue*
# syntax: string (acl)
#tcpdeny=
#
#####################################################################
#            AUTOBAUD function
#
# Some PC (mostly Dell apparently) have a serial console that
# uses some VT100 control parameters to figure out if the
# console is connected, and if so what betrayer to use.
#
# The problem with those consoles is that if no client is connected
# to the line when the host is booted, the serial console does not
```

```
# get any answer to its vt100 requests and the console is not
# becoming active.
#
# By setting the following parameter, ControlTower cans respond
# to the vt100 query string and the serial console on those PC's
# will be activated.
#
# The 3 possible values are
#       - vt100 DC1     respond to DC1
#       - vt100 DC2     respond to DC2
#       - vt100 DS      respond to DS
#attires = vt100 DC1


<==OLD Versions FOl-
lows==================================================>
# ControlTower Console Manager DEFAULT configuration
#
# ******************** DO NOT EDIT THIS FILE *********************
#
# This file is read before the LOCAL file, <group>/<group>.grp files,
# and <system>.cfg files. Any changes to these defaults should be
# made by adding lines to the LOCAL, or per-group configuration files.
#
#########################################################################
# COPYRIGHT (c) 1998 - 2005 BY CARLO GAVAZZI COMPUTING SOLUTIONS.,
BROCKTON, MA.
#
# THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED
AND
# COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND
WITH
# THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR
# ANY OTHER COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE
MADE
```

```
# AVAILABLE TO ANY OTHER PERSON. NO TITLE TO AND OWNERSHIP OF
THE
# PROGRAM IS HEREBY TRANSFERRED.
#
# THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT
# NOTICE AND SHOULD NOT BE CONSIDERED AS A COMMITMENT BY
CARLO GAVAZZI COMPUTING SOLUTIONS.
#
#
#####################################################################
#                  **** NOTE WELL ****
#####################################################################
#
# With very few exceptions, a value for each parameter MUST be
# specified. NO default values are present in the code, so
# values must be specified here, or in a per-system config file.
# All parameters for which a default can be reasonably picked
# appear here.
#
#####################################################################
# serial line parameters
#
# devicename for serial port attached to system console
# syntax: string
#device=
#
# set O_EXCL “exclusive open” bit on serial port open
# syntax: boolean
exclusive=true
#
# create (and honor) UUCP-compatible lock files for the serial port
# syntax: boolean
uulock=true
#
# tty mode.
# syntax: one or more tokens separated by commas or spaces
# tokens;
#integer (speed/baud)
#cs5 cs6 cs7 cs8
```

```
#flag
#-flag
# flags:
#crtscts cstopb parenb parext parodd ixoff ixon istrip
#
# ALL flags/parameters should appear here in DEFAULT file. subsequent
# stty configuration (in <system>.cfg or with -o on command line, or
# via console program “set” command) change ONLY the bits which are
# specified (all other remain the same).
stty=9600 cs8 -crtscts -cstopb -parenb -parext -parodd -ixoff -ixon istrip
#
# allow client programs to change serial line parameters
# syntax: boolean
ttychanges=true
#
# string to send instead of BREAK signal (optional)
# syntax: string
# the following escape sequences are allowed;
#\r \n \t \ooo (1 or more octal digits) \xXX (two hex digits)
#breakstring=
#
######################################################################
#
# Logfile parameters
# All system console output is saved in a logfile.
#
# directory for all log files (must be absolute)
# syntax: string (path)
logdir=/var/log/cmgrlog
#
# log client (user) input in logfile (THIS INCLUDES PASSWORDS!!)
# all output (including echo) is always saved in the logfile
# syntax: boolean
loginput=false
#
# log messages sent to users (user connect/disconnects) in logfile
# (serial line change and break messages are always logged)
# syntax: boolean
```

```
logmessages=true
#
# owner for log files
# syntax: user name or uid
logowner=root
#
# group for log files
# syntax: group name or gid
loggroup=sys
#
# mode for log files
# syntax: octal mode (e.g.; 0600) or comma separated sequence
#of symbolic absolute modes strings [uoga]=[rwxs]+
logmode=u=rw
#
# Optional: strftime(3) format used to timestamp lines in logfile
#if empty, lines are not time-stamped
# syntax: string
loglinestamp=%c
#
#################
# Logfile parameters read by "logcheck" program run every
# 10 minutes from root crontab;
#
# Maximum logfile size in bytes before closing and "rotating";
# syntax: integer
logmaxsize=50000
#
# Number of old log files to compress and keep in "rotation";
# syntax: integer
logmaxfiles=7
#
# Optional: how often to timestamp logfile in minutes; one of: 10, 20, 30, 60
#or zero to disable.
# syntax: integer
logstamp=60
#
# strftime(3) format used to output periodic logfile timestamps;
```

```
#if empty, no periodic timestamps will be output
# syntax: string
logstampformat=********** %c **********
#
# Logfile compression program path
# syntax: string
logcompress=
#
# Logfile compression program options; compression program is expected
# to ALWAYS compress the log file even if this does not result in a
# space savings
# syntax: string
logcompressopt=-f
#
# Logfile compression program output extension (including DOT character)
# syntax: string
logcompressext=.Z
#
# Optional log filter file path. If the path is relative (no leading
# slash), the pathname will be taken as relative to the directory in
# which the device .cfg file was found.
#
# To override a value set in a higher level configuration file, use the
# magic string *novalue*
#
# syntax: string (path)
#logfilter=
#
#####################################################################
#
# Authorization parameters
#
# The name of the user remote users must supply the password for.
# If authfile (below) is specified, authuser only applies to version 1
# viewers.
# syntax: string (user name)
authuser=auracmgr
#
```

```
# The path to an optional per-user authorization file which contains a
# list of authorized users and their capabilities. The path may be
# relative (to group or config directory). If not specified, all
# users must authenticate as "authuser", if set users will be prompted
# for a user name.
#
# To override a value set in a higher level configuration file, use the
# magic string *novalue*
#
# syntax: string (path)
#authfile=
#
##########################################################################
#
# Idle time parameters
#
# If non-zero, the maximum time in minutes before disconnecting an
# idle viewer regardless of whether view is attached/view-only. If
# zero, no idle limit is enforced.
# syntax: integer (minutes)
disconnectidle=0
#
# If non-zero, the maximum time in minutes before detaching (demoting
# to view-only) an idle attached viewer. If zero, no idle limit is
# enforced.
# syntax: integer (minutes)
detachidle=0
#
##########################################################################
# local (Unix domain) socket parameters
#
# Allow local (Unix-domain) connections
# was called unixenable in Version 1.00
# (old name still accepted)
# syntax: boolean
localenable=true
#
# Require password on local (Unix-domain) connections;
```

```
# was called unixenable in Version 1.00
# (old name still accepted)
# syntax: boolean
localauth=false
#
######################################################################
# TCP socket parameters
#
# allow TCP/IP connections
# syntax: boolean
tcpenable=true
#
# The following have no default value, and may be left blank.
# syntax: comma separated list of host/mask pairs.
#
# The host may be a name (from /etc/hosts or DNS) or dotted decimal octets
# (nnn.nnn.nnn.nnn). The mask is optional, and can be used to specify
# which bits of the host address are to be examined.
#
# 1) a single decimal number (/24) signifying the number of high-order
#bits set in the mask
# 2) four dotted decimal octets (/255.255.255.0)
# 3) a single hexadecimal value (/0xffffff00)
#
# if no mask is supplied, all bits in the host address are examined,
# so "host/32" is the same as "host"
#
# If set a host must match an entry in "tcpallow" in order to be accepted.
# To override a value set in a higher level configuration file, use the
# magic string *novalue*
# syntax: string (acl)
#tcpallow=
#
# If set a host must NOT match an entry in "tcpdeny" in order to be accepted
# To override a value set in a higher level configuration file, use the
# magic string *novalue*
# syntax: string (acl)
#tcpdeny=
```

CARLO GAVAZZI
COMPUTING SOLUTIONS

Appendix C

An Example Configuration

This is an example configuration with devices at various levels and in groups and with parameters changed from their defaults at various levels. This configuration has been set up and run in a lab.

Devices have been configured at the top level in the `/opt/AURAcmgr/config` directory and under a group directory. The files and directories under `/opt/AURAcmgr/config` are:

- DEFAULT
- LOCAL
- LOCALauthfile
- device2.cfg
- device3.cfg
- group/
- license.dat

Under the `group` directory are the files:

- device0*
- device0.1.test*
- device0.cfg
- device1.cfg

- group.grp
- groupauthfile

Modifications to global parameters have been made in the LOCAL file instead of the DEFAULT file. The LOCAL file is:

```
# Don't allow client programs to change serial line parameters
ttychanges=false
#
# So no one can send a 'break' to a device, set the
# breakstring to the text 'NO!'.
breakstring=\116\117\041
#
# Everyone can read the log files.
logmode=u=rw,g=r,o=r
#
# Set loglinestamp to NULL so that lines are not time-stamped.
loglinestamp=
#
# Since there are no timestamps per line, there will be timestamps every 10
# min.
logstamp=10
#
# Add the week # to the default periodic time stamp.
logstampformat=********** %c Week#: %U **********
#
# Logfile compression program path
# syntax: string
# Change compression utility to gzip for better compression.
logcompress=/usr/bin/gzip
#
# This is the same as in the DEFAULT file but is left here for clarity.
logcompressopt=-f
#
```

```
# This is the extension that is appended by gzip.
logcompressext=.gz
#
# Put logfilter file with logfiles it affects. This file will filter input to
all log files,
# including ones under a group
logfilter=/var/log/cmgrlog/LOCALlogfilterfile
#
# "Unset" the authuser parameter
authuser=
#
# This file currently contains 'auracmgr all' so that only people with the
# auracmgr
# password have access to devices.
authfile=LOCALauthfile
#
# Set so that a viewer will disconnect if it's been idle for 5 minutes.
disconnectidle=5
#
# Set so that a viewer will detach if it's been idle for 10 minutes.
detachidle=10
#
# Require authorization for even local connections to devices.
localauth=true
#
# Keep out the people in Marketing
tcpdeny=100.100.100.100/8
```

Notice that `authfile` is set to `LOCALauthfile` with no specified path. This is why a `LOCALauthfile` file has been created in the `/opt/AURAcmgr/config` directory. The contents of this file are:

```
auracmgr all
```

The contents of the two device configuration files in the `/opt/AURAcmgr/config` directory are:

- device2.cfg:
- device=/dev/cua/2
- device3.cfg:
- device=/dev/cua/3

The `group.grp` group configuration file under the group directory contains:

```
# The devices in this group belong to development so we will be
# giving more people access to these devices.
authfile=groupauthfile
```

This is why a `groupauthfile` file has been created under the group directory. Its contents are:

```
# As developers are hired, add them to this file.
auracmgr all
developer1 all-break
developer2 attach+fattach+stty
```

The configuration file for device0 under the group directory contains:

```
#
# allow client programs to change serial line parameters
ttychanges=true
#
# In case of problems, we want to be able to send a real break.
breakstring=
#
# This is the directory containing configuration files. The log files will be
# sent here for easy access.
logdir=/opt/AURAcmgr/config
#
# This will write everything typed into a cmgr session, including PASSWORDS!
loginput=true
#
# logmessages=true in the DEFAULT file.
```

```
#
# mode for log files
# Results in -rwxrw-r--
logmode=764
#
# To make debugging easier, line-by-line time stamping is being turned back
# on with the default value.
loglinestamp=%c
#
# This will keep larger log files and more of them.
logmaxsize=500000
#
logmaxfiles=20
#
# Since line time-stamping is back, we don't need as many periodic time
# stamps.
logstamp=60
#
# Return the format of the periodic time stamp to the default.
logstampformat=********** %c **********
#
# Logfile compression program path
# Use compress so that log files can be copied to, and uncompressed on,
# machines that don't have gzip.
logcompress=/usr/bin/compress
#
# This still hasn't changed but is left for clarity.
logcompressopt=-f
#
# Logfile compression program output extension (including DOT character)
# The compress extension is .Z
logcompressext=.Z
#
```

```
# Turn off log filtering.
logfilter=*novalue*
#
# Set authuser to the user whose password everyone has.
authuser=auracmgr
#
# Turn off per-user authorization.
authfile=*novalue*
#
# Turn off idle disconnect.
disconnectidle=0
#
# Turn off idle detach.
detachidle=0
#
# Don't require authorization for local connects.
localauth=false
#
# Turn off tcpdeny so that everyone can look at managed devices.
tcpdeny=*novalue*
#
# Turn on errlog so that error messages will be written to somewhere
# other than syslog.
errlog=device0errlog
#
# Turn on debugging for as much info as possible.
debug=1
```

The settings in this file have been chosen to maximize the information written to log files, including the additional log file, `device0errlog`, which is written to the directory in which the ControlTower server was started, in this case, `/opt/AURAcmgr/config`. Notice that the value of `logdir` has been changed to `/opt/AURAcmgr/config`. This is why there are `device0` log files, `device0` and `device0.1.Z`, in the `/opt/AURAcmgr/config` directory listing above. These files are listed with a star

after the name indicating that these are executable files because logmode was set to 764 (*u=rwx*, *g=rw*, *o=r*).

Notice also that `authuser` is set to `auracmgr` for this device.

The default log directory of `/var/log/cmgrlog`, contains files and directories:

```
device2
device2.1.gz
device2.2.gz
device3
device3.1.gz
group/
```

Under the group directory is:

```
device1
device1.1.gz
device1.2.gz
```

`device1` is the device under the group directory that has no parameter changes of its own.

Glossary

Terms & Definitions

Attach

See Read/Write mode.

AURAcmgr

The ControlTower command line Viewer Client software package. Required for installation of AURAcmgrs package.

AURAcmgrs

The ControlTower Server Software package. Requires that the AURAcmgr package be installed. AURAcmgrs is required for installation of the AURA-jcmgr package.

Break Signal

An RS-232 signal that for some managed devices is interpreted as a device reset command.

Breakout Box

Hardware used to connect RS-232 serial devices to multiport serial cards. Also referred to as a Connection Box.

Character Oriented Viewer Client

Software supplied by Carlo Gavazzi Computing Solutions that provides access to the managed devices' console serial port through a character-oriented window. Also known as "CLI Viewer Client" and "Command Line Viewer Client".

CLI Viewer Client

See Character Oriented Viewer Client.

cmgr

Carlo Gavazzi Computing Solutions supplied software program running the Character Oriented Viewer Client functionality in the existing terminal window.

Command Line Viewer Client

See Character Oriented Viewer Client.

Connect

The act of connecting to a managed device.

Connection Box

See Breakout Box.

Console Management

See Console Management Services.

Console Management Services

Logging and real time viewing of output from managed devices, and control of managed devices.

Console Serial Port

The serial port on the managed device whereby commands can be sent and data received. Also known as “Serial Console Port” and “Console Port”.

Control

See Read/Write mode.

ControlTower Host

See ControlTower Host Server System

ControlTower Host Server System

The computer on which the ControlTower Server Software has been installed, regardless of whether any Aurora brand hardware is installed.

ControlTower Host System

The computer system including: Breakout Boxes, Multiport Serial Cards, Expansion Chassis, and ControlTower Software. Also known as “Host”, “Host System”

ControlTower Server Software

Software supplied by Carlo Gavazzi Computing Solutions that provides console management services, see the entries for AURAcmgrs, AURAjcmgr.

Force Control

See Force Read/Write mode.

Force Read/Write Mode

The ability of the Viewer Client to take Read/Write mode if there is already another user that has Read/Write mode.

Host

See ControlTower Host Server System.

Host System

See ControlTower Host Server System.

Local Viewer Client

A Character Oriented Viewer Client that is run directly on the ControlTower Host, without the CONSOLE_SERVER environment variable defined.

Log

See Log File.

Log File

Output from a managed device that is stored locally on the ControlTower Host.

Managed Device

A computer or other system that accepts basic management commands over an RS-232 serial interface; see Console Serial Port.

Monitor

See Read-Only mode.

Network Client

GUI or CLI connection to ControlTower Host using TCP/IP.

Package

A Solaris software package that is installed on a computer system using the Solaris system command, pkgadd. Package removal is done with the Solaris system command, pkgrm.

Read-Only mode

The ability of the Viewer Client to monitor output from the managed device. A Viewer Client connection that allows the user to view all managed device output as it happens, but not to send any keystrokes to the managed device. Requires the “connect” capability in the managed system authfile.

Read/Write mode

The ability of the Viewer Client to interact with the managed device. A Viewer Client connection that allows the user to see all managed device output as it happens, and to send keystrokes to the managed device. The act of entering read/write mode is called “attaching”, and requires the “attach” capability in the managed system authfile. If another user is currently attached (in read/write mode), you can forcibly take control away from them (this requires both the “attach” and “fattach” capabilities in the managed system authfile).

View

See Read-Only mode.

Viewer Client

Carlo Gavazzi Computing Solutions supplied software that provides ability to issue commands to a managed device’s console serial port, view log files and interact with the ControlTower Server Software.

Index